

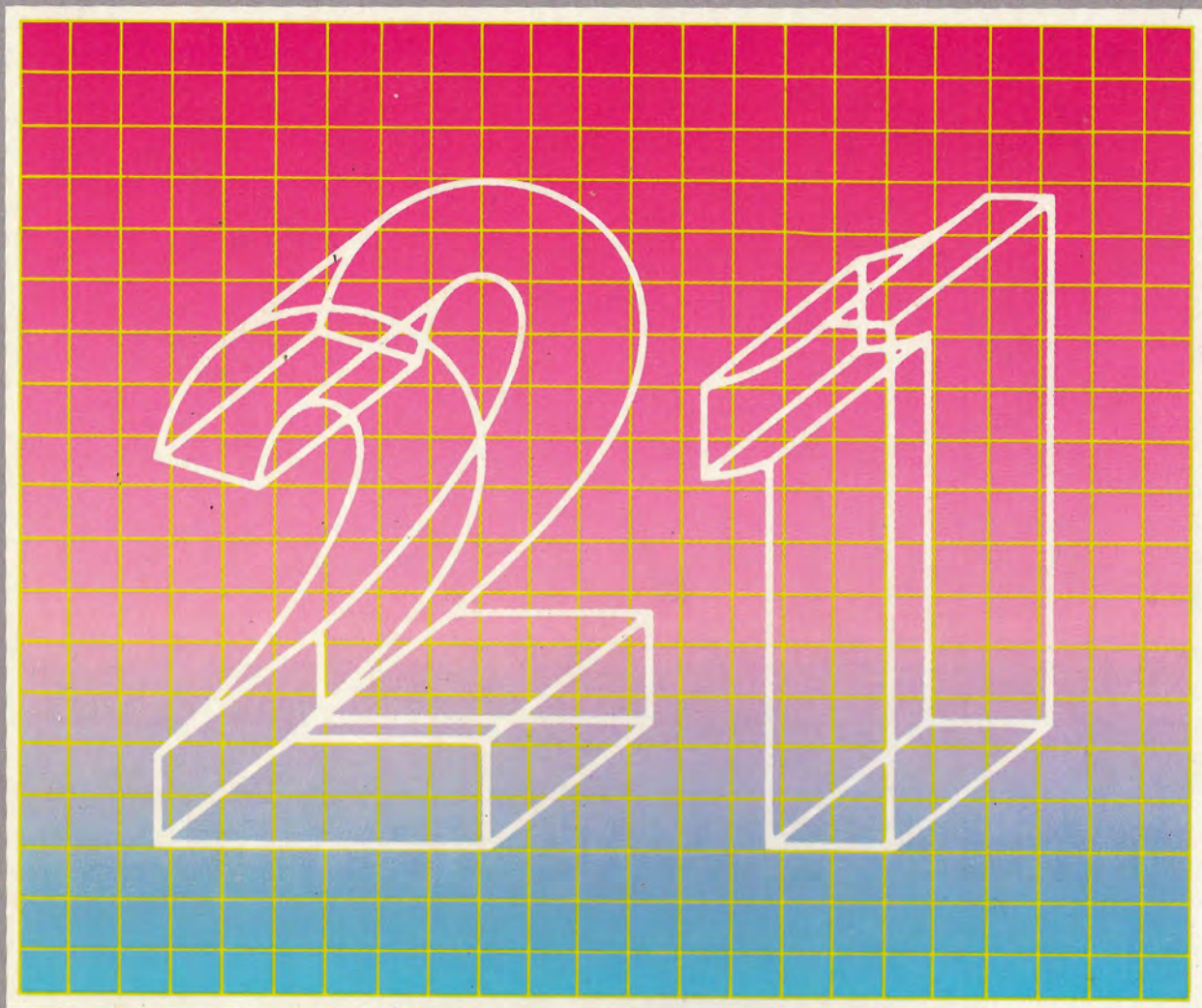
BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,

IBM, SPECTRUM, COMMODORE Y MSX



¿QUIERE APRENDER INFORMATICA?

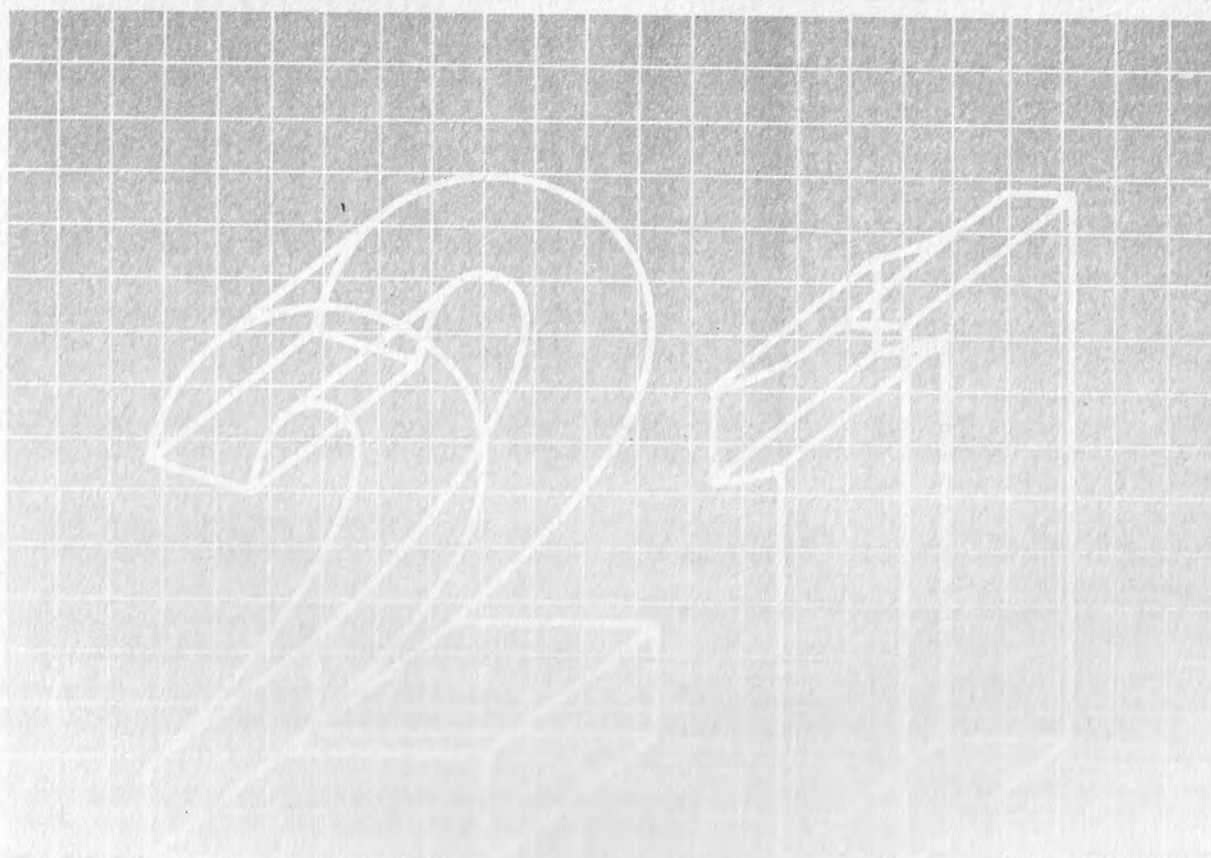
PRACTIQUE CON SU ORDENADOR

EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 21. «Experiencias y prácticas en logo», Carlos Albert, Técnico de Informática; Adoración Llena Jubero, Técnico de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: José María Esteban, Diplomado en Telecomunicación; Alejandro Marcos, Licenciado en Química. «El Taller del Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-124-X

ISBN de la obra: 84-7688-047-2

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

MATEU CROMO. PINTO (Madrid).

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Junio, 1987

P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	15
■ TRUCOS Y RUTINAS BASICAS	21
■ EL TALLER DEL HARDWARE	29
■ APRENDER CON EL ORDENADOR	39
■ PEQUEÑA HISTORIA DE LA INFORMATICA	49
■ TEMAS MONOGRAFICOS DE VANGUARDIA	53
■ TERMINOLOGIA	57
■ VOCABULARIO DE INFORMATICA	59

GUARDAR Y RECUPERAR DIBUJOS

■ Guardar dibujos



A conocer la forma de guardar procedimientos en diferentes soportes de información, que dependiendo del ordenador, consiste en uno u otro. La forma de realizarlo, también varía según el ordenador con el que tra-

bajamos.

El Logo nos permite también guardar directamente dibujos que hayamos realizado.

En este caso lo que se guarda no es un procedimiento que realiza un dibujo determinado, sino un dibujo que se encuentra en la pantalla.

La primitiva que nos permite esto es:

— GUARDADIB

Esta orden es la abreviatura de GUARDADIBujo.

La forma de utilizarla es:

? GUARDADIB "nombre fichero

Una vez que esta orden es introducida, el contenido que haya en ese momento en la pantalla será almacenado en el fichero que se especifica.

En la pantalla también se encuentra texto, éste será también almacenado.

También vimos que cuando los procedimientos se guardaban, el nombre del fichero que habíamos dado aparecía con una extensión, que consistía en un punto y tres letras (.LOG). De esta forma sabíamos que el fichero

correspondía a un fichero del Logo compuesto de procedimientos.

A la hora de guardar dibujos ocurre lo mismo, pero en este caso la extensión varía y es:

.DIB

De esta forma siempre que nos encontremos un nombre de fichero con la extensión .DIB, sabremos que se trata de un dibujo que hemos guardado previamente.

También podemos encontrar versiones que en lugar de la extensión .DIB, ponen la extensión .IM.

La primitiva GUARDADIB corresponde a las versiones del Logo para los MSX y PC-COMPATIBLES.

Para el caso del SPECTRUM es:

? GUARDAPANT "nombre fichero

Abreviatura de GUARDAPANTalla.

■ Recuperar dibujos

Al igual que los procedimientos que previamente se habían guardado se podían recuperar, es decir, volver a cargarlos en la memoria del ordenador, las pantallas guardadas también pueden ser recuperadas.

Para esto lo único que hay que hacer es dar la orden oportuna para cargar de nuevo la pantalla deseada.

La orden que nos permite esto es:

— CARGADIB

Abreviatura de CARGADIBujo.

Se puede guardar el contenido de la pantalla en un fichero.

EXPERIENCIA Y PRACTICAS EN LOGO

La forma de dar correctamente esta orden es:

? CARGADIB "nombre fichero

Una vez introducida, el fichero que se especifica se recuperará en la pantalla.

El fichero que se especifica tiene que corresponder con uno que previamente se haya guardado con GUARDADIB.

En la pantalla aparecerá una copia exacta de lo que se había guardado, tanto gráficos como textos.

El archivo que se carga no podrá ser nunca uno con la extensión .LOG.

Al igual que ocurría con GUARDADIB, la primitiva CARDADIB es válida para las versiones de los MSX y PC-COMPATIBLES, pero para las versiones de Logo del SPECTRUM esta orden es:

? CARGAPANT "nombre fichero

Abreviatura de CARGAPANTalla.

Es conveniente que la pantalla esté en modo mixto o gráfico para que la recuperación del fichero sea correcta.

■ La recursividad

Recordemos con unos ejemplos la parte que ya conocemos de la recursividad.

— Recursividad en un procedimiento sin entradas.

? PARA CIR

> MT

> REPITE 36 [AV 5 GD 10]

> GD 45

> CIR

> FIN

La Tortuga permanecerá dibujando círculos de igual tamaño variando la posición 45 grados, hasta que detengamos manualmente la ejecución del procedimiento.

? BP CIR

— Recursividad en un procedimiento con una entrada.

? PARA ECIR :T

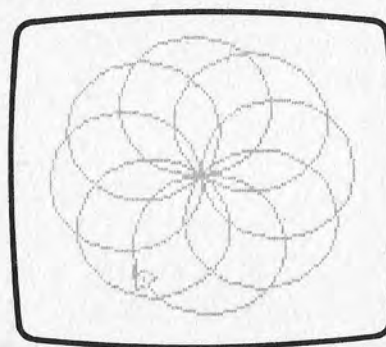


Fig. 1.

```
> MT
> REPITE 36 [AV :T GD 10]
> GD 45.
> ECIR (:T + 0,1)
> FIN
```

El procedimiento ECIR tiene una entrada que determina el tamaño del círculo, en la llamada recursiva incrementamos el tamaño del círculo 0,1 puntos.

La ejecución del procedimiento también debe detenerse manualmente.

Así pues, la única diferencia respecto al procedimiento CIR, es que en éste vamos variando el tamaño de cada círculo que se dibuja.

? ECIR 2

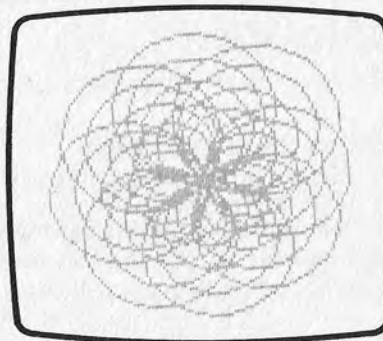


Fig. 2.

El tamaño del primer círculo que se dibuja lo determinamos cuando ejecutamos el procedimiento.

Utilizando procedimientos recursivos podemos obtener bonitos dibujos.

Pero la recursividad es todavía más potente.

Veamos su efecto en procedimientos de dos entradas.

```
? PARA GECIR :T :G
> REPITE 36 [AV :T GD 10]
> GD :G
> GECIR (:T + 0.1)(:G + 45)
> FIN
```

El procedimiento GECIR tiene dos entradas:

T determina el tamaño del círculo.

G determina el giro que da la Tortuga entre un círculo y otro.

En la llamada recursiva variamos los valores de entrada del procedimiento, consiguiendo así que el tamaño del círculo aumente continuamente en 0,1 puntos y que el giro entre un círculo y el siguiente aumente cada vez en 45 grados.

? GECIR 2 45

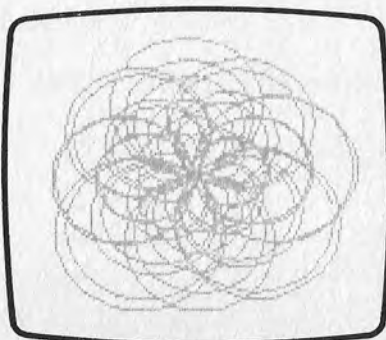


Fig. 3.

Hemos ejecutado el procedimiento inicializando el tamaño del círculo a 2 y el giro a 45.

Y como en los casos anteriores la ejecución se detiene manualmente.

Observando la sintaxis de las llamadas recursivas, en procedimientos de una y dos entradas, deducimos que en un procedimiento de N entradas la llamada recursiva debe contener N datos.

■ Recursivación con parada

La solución a la pesada tarea de detener manualmente la ejecución de un procedimiento recursivo es:

Introducir en el procedimiento una condición de parada.

```
? PARA NCIR :N
> SI :N < 1 [ALTO]
> REPITE 36 [AV 5 GD 10]
> GD 45
> NCIR :N - 1
> FIN
```

Una de las líneas del procedimiento NCIR es una condición; si el valor de la variable N es menor que 1, el procedimiento se detendrá.

La variable N adquiere valor la primera vez que se ejecuta el procedimiento, a partir de entonces ese valor inicial irá decremintándose en una unidad, y cuando sea inferior a uno, la ejecución del procedimiento se detendrá.

Si queremos que la Tortuga dibuje cinco círculos tecleamos:

? NCIR 5

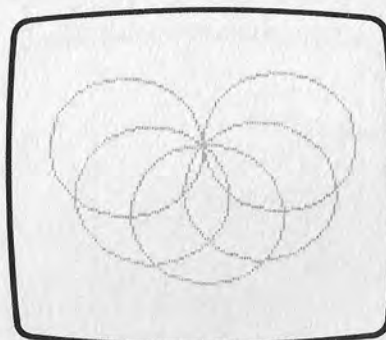


Fig. 4.

si queremos que sean ocho:

? NCIR 8

Con una sola condición hemos indicado el número de veces que queremos que se ejecute el procedimiento recursivo.

Si nos encontramos ante un caso en el que el número de veces que debe ejecutarse

La ejecución de un procedimiento recursivo puede detenerse incluyendo en el procedimiento una condición.

EXPERIENCIA Y PRACTICAS EN LOGO

un procedimiento es desconocido, podemos establecer una salida que dependa de una acción que se realice.

Por ejemplo, que el procedimiento se detenga si se ha pulsado alguna tecla.

```
? PARA CUADRADOS
> SI TECLA? [ALTO]
> REPITE 4 [AV 20 GD 90]
> GD 45
> CUADRADOS
> FIN
```

En CUADRADOS, si se ha pulsado una tecla el procedimiento se detiene, si no la Tortuga dibuja cuadrados iguales en la pantalla.

También podemos establecer que sea una tecla determinada la que dé fin a la ejecución.

```
? PARA CUADRADOS
> SI TECLA? [SI LEECAR= "F" [ALTO]]
> REPITE 4 [AV 30 GD 90]
> GD 20
> CUADRADOS
> FIN
```

El procedimiento CUADRADOS se ejecutará hasta que se pulse la tecla (F).

?CUADRADOS

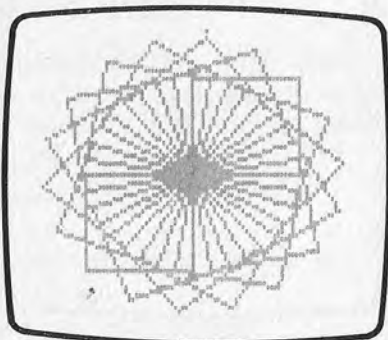


Fig. 5.

Cuando vemos que la figura ya se ha dibujado detenemos la ejecución.

Ahora que ya sabemos mucho sobre la recursividad, repitamos un ejercicio que ya vimos con anterioridad, pero aplicando en él la recursividad.

El ejercicio en cuestión nos permite dibujar, moviendo la Tortuga con las teclas del cursor, directamente sobre la pantalla.

Primero definimos un procedimiento para inicializar la pantalla.

```
? PARA DEFID
> BP
> BL
> PONCL 1
> MT
> HAZ "P 1
> DIBU
> FIN
```

El procedimiento DEFID le pasa el control al procedimiento DIBU.

```
? PARA DIBU
> POSID
> SI :F = 27 [ALTO] [DIBU]
> FIN
```

DIBU se reejecutará mientras no se pulse la tecla ESC.

Cada vez que se ejecuta DIBU se ejecuta POSID.

```
? PARA POSID
> HAZ "B LEECAR
> HAZ "F ASCII :B
> SI :F = 43 [HAZ "P :P + 1 ALTO]
> SI :F = 45 [HAZ "P :P - 1 ALTO]
> SI :F = 66 [BL ALTO]
> SI :F = 83 [SL ALTO]
> SI :F = 71 [GOMA ALTO]
> SI :F = 82 [RELLENA ALTO]
> SI :F = 78 [BP ALTO]
> SI :F = 27 [ALTO]
> HAZ "B LEECAR
> HAZ "F ASCII :B
> SI :F = 72 [PONRUMBO 0 AV :P]
> SI :F = 80 [PONRUMBO 180 AV :P]
> SI :F = 75 [PONRUMBO 270 AV :P]
> SI :F = 77 [PONRUMBO 90 AV :P]
> FIN
```

Las teclas que tienen una función en el manejo de este procedimiento son:

+ Pulsando esta tecla aumentamos el número de puntos que salta la Tortuga al dar un paso.

- Para decrementar el número de puntos de cada paso.

B Activar el lápiz.

S Desactivar el lápiz.

G Activar la goma.

La recursividad simplifica la resolución de procedimientos.

R Rellenar.
N Borrar la pantalla.
ESC Detener la ejecución.

Y las cuatro teclas de movimiento del cursor para dirigir a la Tortuga.

Introduce estos tres procedimientos en tu ordenador e intenta realizar el siguiente dibujo.

Para empezar teclea:

? DEFID

y empieza a dibujar.

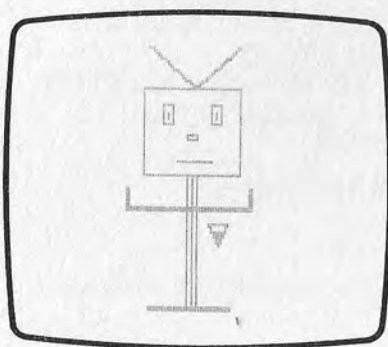


Fig. 6.

Recuerda que puedes guardar un dibujo con la primitiva:

(Dependiendo del ordenador.)

- PC-Compatibles
? GUARDADIB "nombre del fichero
- SPECTRUM
? GUARDAPANT "nombre del fichero

- MSX
? GUARDADIB "nombre del fichero

Y que puedes recuperarlo con la primitiva:

(Dependiendo del ordenador.)

- PC-Compatibles
? CARGADIB "nombre del fichero
- SPECTRUM
? CARGAPANT "nombre del fichero
- MSX
? CARGADIB "nombre del fichero

Un fichero, guardado mediante "GUARDADIB", nunca se puede cargar con "CARGA".

■ Cuadro resumen

- GUARDADIB "nombre de fichero
- GUARDAPANT "nombre de fichero
(Para el SPECTRUM)

El contenido de la Pantalla en el momento que se introduce esta orden (gráficos y/o texto), es almacenado en el soporte activado bajo el nombre del fichero que se especifica.

- CARGADIB "nombre de fichero
- CARGAPANT "nombre de fichero
(Par el SPECTRUM)

El fichero que se especifica se recupera en la pantalla. La pantalla aparece exactamente igual a como estaba en el momento de haber sido guardada.

■ Ejercicios

1. Define un procedimiento que realice el siguiente dibujo, teniendo en cuenta que cuando pulse la tecla (M) la Tortuga debe detenerse unos instantes, y que cuando se pulse la tecla (F) la ejecución del procedimiento debe mantenerse por completo.

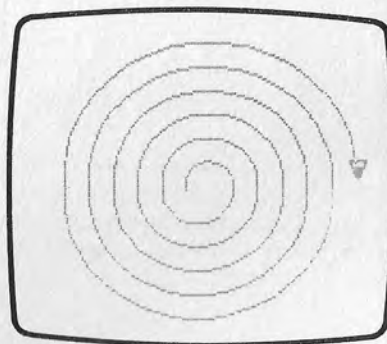


Fig. 7.

2. Utilizando la recursividad, resuelve el problema de hallar el factorial de cualquier número.
3. Intenta realizar el siguiente dibujo:

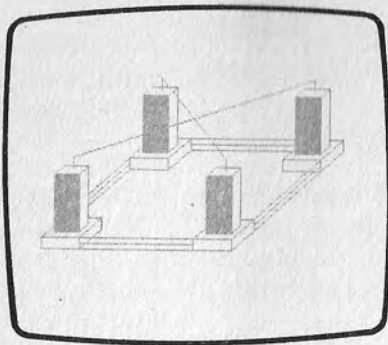


Fig. 8.

4. Define un procedimiento recursivo para cada una de las espirales siguientes:

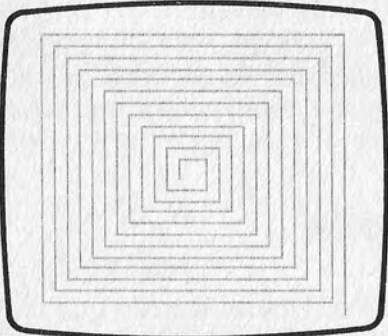


Fig. 9.

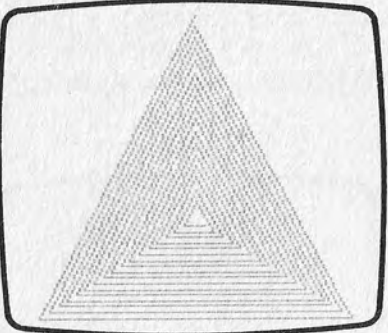


Fig. 10.

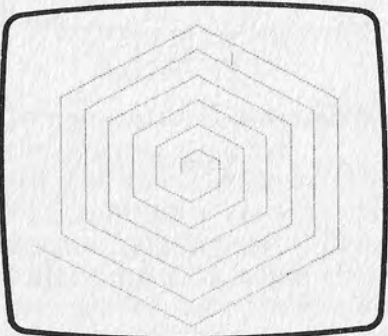


Fig. 11.

Define un único procedimiento con el que se puedan dibujar las tres espirales anteriores, dependiendo de los valores que se den a variables de entrada.

5. ¿Son correctas las órdenes siguientes?

```

— ? PARA A :V
  > AV :V
  > GD :V + 10
  > A
  > FIN

— ? PARA B
  > B
  > FIN

— ? PARA C
  > HAZ "V 10
  > REPITE 4 [AV :V GD 90]
  > C :V + 4
  > FIN

— ? PARA NADA
  > MT
  > OT
  > NADA
  > FIN

— ? PARA MAS
  > SI :L = 0 [ALTO]
  > MAS
  > FIN

— ? PARA CASO :L :A
  > ESCRIBE :L
  > ESCRIBE :A
  > SI :L > 50 [HAZ "L 5]
  > GI :A
  > AV :L
  > SI :A > 300 [ALTO]
  > CASO :L + 1 :A + 1
  > FIN
    
```

Solución de los ejercicios

1:

El procedimiento para realizar el dibujo sería el siguiente:

```

? PARA CURVA :L
  > AV :L
  > REPITE 18 [AV :L GD 10]
  > CURVA :L + 1
  > FIN
    
```

Tanto los textos como los gráficos pueden ser almacenados en un fichero.

Como queremos que al pulsar la tecla (M) la Tortuga se detenga unos instantes, establecemos una condición.

```
SI TECLA? [SI LEECAR = "M [ESPERA 40]]
```

También queremos que la ejecución finalice si se pulsa la tecla (F), lo cual implica otra condición.

```
SI TECLA? [SI LEECAR = F [ALTO]]
```

Una vez establecidas las dos condiciones las añadimos en el procedimiento anterior, y nos queda:

```
? PARA CURVA :L
> AV :L
> REPITE 18 [AV :L GD 10]
> SI TECLA? [SI LEECAR = "M [ESPERA 40]]
> SI TECLA? [SI LEECAR = "F [ALTO]]
> CURVA :L + 1
> FIN
```

Para ejecutarlo tecleamos

```
? CURVA 2
```

y comprobamos que funciona todo bien pulsando las teclas (M) y (F).

2:

La función principal del procedimiento FAC es inicializar la variable en la que se acumularán los productos.

La entrada de este procedimiento será el número del que queremos hallar el factorial.

Una vez inicializada la variable C, llamamos al procedimiento FACT indicando los datos M y C.

M número
C valor inicial del acumulador

```
? PARA FAC :M
> HAZ C (:C * :M)
> FACT :M :C
> FIN
```

En FACT se calcula y escribe el factorial de M.

```
? PARA FACT :M :C
? HAZ C (:C * :M)
```

```
> SI :M = 1 [ESCRIBE :C ALTO]
> FACT :M - 1 :C
> FIN
```

Para calcular el factorial de 5

```
? FACT 5
120
```

3:

Para realizar el dibujo vamos a ir definiendo procedimientos que dibujen diferentes partes.

En primer lugar definimos el que dibuja la base de las columnas.

Utilizamos dos variables, PX y PY, para determinar el punto de la pantalla donde va a ir dibujada cada una de las columnas. Desde ese punto (PX, PY), se comienza a dibujar la base:

```
? PARA BASE :PX :PY
> SL
> PONX :PX
> PONY :PY
> PONRUMBO 0
> BL
> REPITE 2 [AV 10 GD 90 AV 30 GD 90]
> AV 10 GD 45
> AV 13
> SL
> AV 12 GD 45
> AV 20
> BL
> AV 10 GD 90
> REPITE 2 [AV 10 GD 45 AV 25 GD 135]
> SL
> PONX :PX + 10
> PONY :PY + 15
> PONRUMBO 0
> BL
> COLUMNA
> FIN
```

Sólo dibujamos la parte de la base que se ve. Las líneas que van a aparecer tapadas no las dibujamos.

Este procedimiento hace al final una llamada a COLUMNA que es el que va a dibujar

Para recuperar un fichero correctamente en pantalla es conveniente estar en modo mixto o de gráficos.

EXPERIENCIA Y PRACTICAS EN LOGO

la columna. Previamente, colocamos a la Tortuga en el punto adecuado de la base:

```
? PARA COLUMNA
> REPITE 2 [AV 50 GD 90 AV 20 GD 90]
> AV 50 GD 45
> AV 10 GD 45
> AV 20 GD 90
> REPITE 2 [AV 50 GD 45 AV 10 GD
135]
> GD 45
> SL
> AV 19
> BL
> RELLENA
> FIN
```

Si ahora ejecutamos el procedimiento BASE con los dos valores siguientes:

```
? BASE 0 0
```

obtenemos el siguiente dibujo:

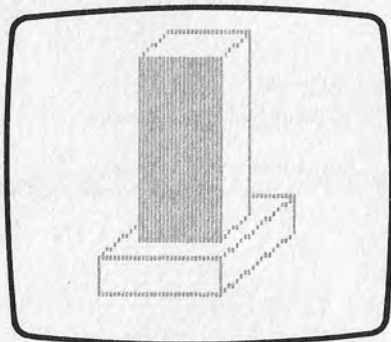


Fig. 12.

A continuación definimos un procedimiento para dibujar las cuatro columnas en la pantalla. Para ello lo que hacemos es cuatro llamadas a BASE con diferentes valores de entrada:

```
? PARA CUATRO
> BASE -100 -60
> BASE 20 -60
> BASE -30 10
> BASE 90 10
> FIN
```

Una vez tenemos las cuatro columnas dibujadas, definimos el procedimiento que dibuja las líneas inferiores que las rodean:

```
? PARA LINEA
> BL
> PONRUMBO 45
> SL
> AV 53
```

```
> BL
> AV 46
> SL
> AV 25 GD 45
> AV 30
> BL
> AV 90
> SL
> AV 30 SL 135
> SL
> AV 25
> BL
> AV 74
> SL
> AV 25 GD 45
> AV 30
> BL
> AV 90
> FIN
```

El procedimiento LINEA1, dibuja las líneas que va de columna a columna:

```
? PARA LINEA1
> PONRUMBO 0
> SL
> PONPOS [-75 10]
> BL
> AV 10
> PONPOS [115 90]
> RE 10
> SL
> PONPOS [45 10]
> BL
> AV 10
> PONPOS [-5 90]
> RE 10
> FIN
```

Una vez que tenemos las diferentes partes del dibujo, definimos un último procedimiento que una a todos los demás:

```
? PARA DIBUJO
> BP
> OT
> CUATRO
> SL
> PONPOS [-100 -60]
> LINEA
> SL
> PONPOS [-100 -55]
> LINEA
> SL
> PONPOS [-100 -50]
> LINEA
> LINEA1
> FIN
```

Para ejecutarlo:

? DIBUJO

4:

Para dibujar este tipo de espirales tenemos que ir aumentando el tamaño de los lados que la forman. Para ello definimos procedimientos recursivos con una variable de entrada que es la que va a determinar la longitud del primer lado. La llamada de nuevo procedimiento se realiza asignando un valor a la variable aumentando en una cantidad fija.

También tenemos que fijarnos en el ángulo que están formando los lados que la componen:

La primera, tiene sus lados formando ángulos de 90 grados.

El procedimiento que la dibuja es:

```
? PARA A :L
> AV :L
> GD 90
> A :L + 3
> FIN
```

Hemos fijado el aumento del tamaño de cada lado, en tres puntos.

Para ejecutarlo bastará con indicar el nombre del procedimiento y asignar un valor inicial a la variable, por ejemplo:

? A 10

La segunda espiral es triangular, y sus lados están formando ángulos de 120 grados:

```
? PARA B :L
> AV :L
> GD 120
> B :L + 3
> FIN
```

Al igual que la anterior para ejecutarlo podemos introducir, por ejemplo:

? B 10

Por último, tenemos una espiral con forma de pentágono, así que sus ángulos tienen que ser de 72 grados:

```
? PARA C :L
> AV :L
> GD 72
> C :L + 3
> FIN
```

Para ejecutarlo:

? C 10

Observa que siguiendo esta regla podemos dibujar, con sólo ir variando los grados, espirales con todas las formas de las figuras geométricas.

Por ejemplo, si queremos una con forma de hexágono:

```
? PARA D :L
> AV :L
> GD 60
> D :L + 3
> FIN
```

Si lo ejecutamos:

? D 10

obtenemos:

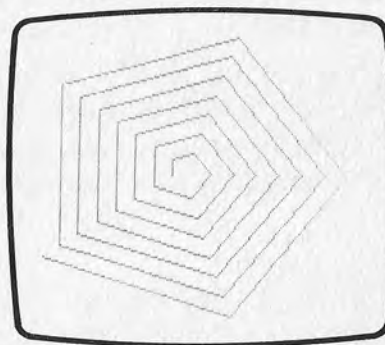


Fig. 13.

Para dibujar, aparte de las espirales anteriores, todas las que queramos, tenemos que definir un procedimiento con dos variables de entrada. Una para el lado y otra para el ángulo.

A la hora de realizar la llamada a sí mismo, tenemos que ir aumentando el valor de la variable que determina la longitud del lado, pero dejar fijo el valor del ángulo:

```
? PARA ESPIRALES :L :A
> AV :L
> GD :A
> ESPIRALES :L + 3 :A
> FIN
```

Si ejecutas este procedimiento con diferentes valores, puedes obtener espirales como ésta:

? ESPIRALES 10 122

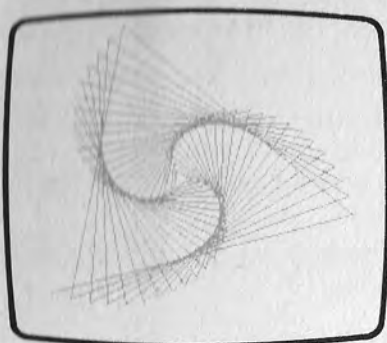


Fig. 14.

5:

```
? PARA A :V
> AV :V
> GD :V + 10
> A
> FIN
```

INCORRECTO

Hay que dar un valor a la variable :V cuando se vuelve a realizar una llamada al procedimiento A.

```
— ? PARA B
> B
> FIN
```

CORRECTO.

El procedimiento B no realiza ninguna acción y estaría indefinidamente ejecutándose.

```
? PARA C
> HAZ "V 10
> REPITE 4 [AV :V GD 90]
> C :V + 4
> FIN.
```

La llamada que realiza el procedimiento C no puede llevar datos para una variable, ya que este procedimiento no dispone de variable de entrada.

```
— ? PARA NADA
> MT
> OT
> NADA
> FIN
```

CORRECTO

El procedimiento NADA mantiene a la Tortuga parpadeando continuamente.

```
— ? PARA MAS
> SI :L = 0 [ALTO]
> MAS
> FIN
```

INCORRECTO

Si ejecutamos el procedimiento MAS el LOG mostrará el mensaje de error:

:L NO TIENE VALOR EN MAS

Recuerda que el LOGO exige que las variables se inicialicen.

```
— ? PARA CASO :L :A
> SI :L > 50 [HAZ "L 5]
> GI :A
> AV :L
> SI :A > '300 [ALTO]
> CASO (:L + 1) (:A + 1)
> FIN
```

CORRECTO

Este procedimiento tiene dos entradas, por tanto, en la llamada recursiva indicamos dos datos.

El procedimiento caso se ejecutará mientras que el valor de la variable A sea menor o igual que 300.

La extensión de un fichero, donde se ha almacenado una pantalla, es diferente a la extensión de un fichero que contiene procedimientos.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS



A hemos visto todo lo que podíamos ver sobre la baja resolución de nuestro ordenador. Ya sabemos cómo definir caracteres y SPRITES. Sabemos cómo diseñar figuras sencillas y complejas, cómo moverlas en la pantalla en cualquier dirección. También sabemos cómo realizar movimientos simultáneos y calcular los movimientos del enemigo. Hemos visto cómo utilizar el teclado para controlar el movimiento y cómo realizar tablas de records para incluirlas en nuestros juegos. Hemos visto todo lo que necesitamos saber para realizar nuestro primer juego con nuestro ordenador. Pero si empezamos ahora con el diseño del juego, nos va a quedar un poco aburrido, pues no hemos hablado para nada del color, de sus posibilidades y de su utilización. Veremos ahora cómo utilizar el color en nuestros juegos.

Utilización del color

Hemos insistido mucho en la buena presentación de nuestros programas, por lo que no podía faltar una explicación sobre el uso e importancia del color.

El uso del color en nuestros programas (aunque éstos no sean de juegos ni gráficos) es realmente importante. Un programa que no tenga cierto colorido para el usuario nunca podrá ser un éxito y, con seguridad, terminará arrinconado y el usuario se olvidará de que existe. Por ello, tenemos que aprender a utilizarlo. Pero no vale utilizarlo sin ton ni son. Tan malo y desagradable puede resultar un programa sin colores como un programa con demasiados colores o colores mal organizados.

Por ello, también veremos cuál es la mejor forma de utilizar el color para que no resulte excesivo.

Lo primero que vamos a ver son las capacidades de color que tienen cada uno de los ordenadores y cómo aprovecharlas. Como cada ordenador tiene una forma distinta de tratar y utilizar el color, vamos a separar la explicación según el ordenador. Más adelante, cuando veamos las reglas básicas para su utilización de forma que los colores no molesten al usuario, daremos una serie de consejos que, salvo excepciones, servirán para todos los ordenadores.

El color en el SPECTRUM

El SPECTRUM fue, junto con el COM-MODORE, el primer ordenador doméstico que apareció en el mercado con la posibilidad de usar colores. Aunque no se puede decir que disponga de muchos y que su utilización sea muy cómoda, éstos son suficientes para un programador medio.

El SPECTRUM tiene 8 colores. Estos son:

0. Negro
1. Azul oscuro
2. Rojo
3. Morado o magenta
4. Verde
5. Azul claro o cian
6. Amarillo
7. Blanco

El ordenador asigna un número a cada color. Dicho número es el que hemos puesto junto a cada color. Cuando tengamos que decirle al ordenador que queremos cierto color, tendremos que darle el número de dicho color y no sus nombres.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

100 REM *****
110 REM * PROGRAMA PARA VER LOS OCHO *
120 REM * COLORES DE QUE DISPONE EL *
130 REM * SPECTRUM. *
140 REM *****
150 REM
160 CLS
170 DIM A$(96)
180 FOR I=0 TO 6
190   INK I
200   PRINT A$
210 NEXT I
220 PAUSE 0

```

Con el primer programa podremos ver los ocho colores que hay en nuestro SPECTRUM.

Todos estos colores, excepto el negro, pueden tener dos niveles de brillo. El brillo lo apreciaremos en la pantalla de nuestro ordenador porque los colores aparecen más claros.

```

100 REM *****
110 REM * PROGRAMA PARA VER EL BRILLO *
120 REM * DE LOS DISTINTOS COLORES EN *
130 REM * EL SPECTRUM. *
140 REM *****
150 REM
160 CLS
170 DIM A$(48)
180 FOR I=0 TO 6
190   INK I
200   BRIGHT 0
210   PRINT A$;
220   BRIGHT 1
230   PRINT A$
240 NEXT I
250 INK 7
260 BRIGHT 0
270 PRINT #0,A$;
280 BRIGHT 1
290 PRINT #0,A$
300 PAUSE 0

```

El segundo programa nos permite ver la diferencia entre un color sin brillo y el mismo color con brillo. Se puede apreciar que el color cero (NEGRO) no cambia cuando se dibuja con brillo. Según esto, podemos decir que el SPECTRUM no tiene ocho colores sino quince.

Antes de continuar tenemos que dejar claros un par de conceptos. Lo primero que observamos al mirar la pantalla de nuestro televisor, es que está dividida en dos zonas. La primera de ellas es la pantalla propiamente di-

cha, y será en ella donde ocurran todas las cosas que veamos. La otra parte rodea a la pantalla como si fuese un anillo. A esta segunda parte la llamaremos BORDE.

Como ya hemos dicho, todo lo que nosotros le digamos al ordenador que imprima, lo realizará en la pantalla. Nunca nos imprimirá nada en el borde de la pantalla. Por eso vamos a ver cómo nos imprime el ordenador las cosas.

Cuando encendemos el ordenador, la pantalla está de color blanco (el borde también) y todo lo que escribamos aparece en negro. Según esto, podemos imaginarnos que la pantalla fuese un papel y las letras estuviesen escritas con tinta o con lápiz. Si nos imaginamos esto con facilidad, también nos podemos imaginar que los colores del papel y de la tinta pueden cambiar. Esto será justamente lo que haremos.

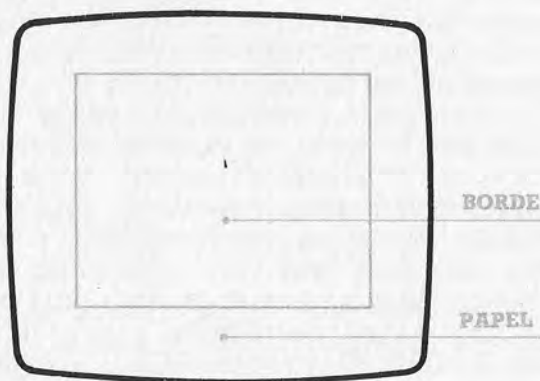


Fig. 1.

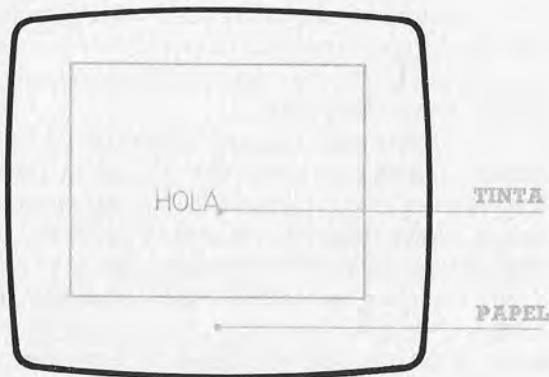


Fig. 2.

También es lógico pensar que podemos cambiar el color del papel sin tener que cambiar el color de la tinta y que podemos cambiar el color de la tinta sin tener que cambiar el color del papel. Por eso, el BASIC del SPECTRUM nos proporciona dos comandos distintos para escribir con colores. Uno de ellos afectará sólo al papel y el otro sólo a la tinta. Estos dos comandos son:

PAPER n. Nos permite cambiar el color del papel al color número «n». El valor de «n» puede estar comprendido entre 0 y 7. Ya veremos más adelante cómo hacer que el color brille.

INK n. Nos permite cambiar el color de la tinta al color «n». Como nota diremos que TINTA en inglés se escribe INK.

No nos hemos olvidado del borde de la pantalla. Aunque no podamos escribir en ella, es lógico pensar que también la podemos cambiar de color para que no estropee la forma ni la presentación de nuestros programas. Para cambiar el color del borde, disponemos del siguiente comando:

BORDER n. Nos permite cambiar el color del borde al color «n». El borde no puede tener brillo, por lo que los colores siempre aparecerán con poca intensidad.

```

100 REM *****
110 REM * PROGRAMA PARA VER COMO *
120 REM * CAMBIA DE COLOR EL BORDE *
130 REM *****
140 REM
150 FOR I=0 TO 7
160   BORDER I
170   FOR J=1 TO 100
180     NEXT J
190 NEXT I

```

Antes de continuar con el brillo vamos a entender un poco mejor cómo se utiliza y cómo funciona el color. Se recordará que la pantalla del SPECTRUM estaba formada por 22 líneas de 32 caracteres cada línea. Por lógica, cada posición de carácter en la pantalla puede tener un color de papel y un color de tinta distinto de los demás. Incluso puede tener un brillo distinto.

Esto nos permite poder cambiar el color del papel y de la tinta de cada carácter. Para realizarlo no hace falta hacer nada en especial. El ordenador está hecho de tal forma que, cuando cambiamos la tinta o el papel, lo

que ya está escrito permanece del color que tenga, y el nuevo valor de tinta o papel será válido sólo para las siguientes instrucciones PRINT que utilicemos. Gracias a esta particularidad hemos podido realizar los programas uno y dos, pues, si cada vez que cambiásemos la tinta, cambiasen de color todos los caracteres de la pantalla no hubiésemos podido ver todos los colores juntos.

Pero puede resultar realmente aburrido (y lento) el intentar escribir cada letra de un color utilizando las sentencias INK y PAPER. Podemos hacer que el color cambie de dos formas mucho más sencillas. La primera de ellas es incluir la instrucción de color dentro de la propia instrucción PRINT. Por ejemplo, si queremos que la palabra «HOLA» aparezca con papel negro y letras en amarillo, no tenemos que hacer lo siguiente:

PAPER 0:INK 6:PRINT "HOLA"

podemos escribirlo de esta otra manera:

PRINT PAPER 0;INK 6;"HOLA"

Lo único que tenemos que tener en cuenta es que todas las instrucciones de color han de ir separadas unas de otras (y de los demás argumentos por la sentencia PRINT) mediante un punto y coma (;).

La otra forma de incluir el color dentro de las sentencias PRINT es mucho más sencilla y rápida. Consiste en cambiar el color del mensaje a imprimir de forma que, dentro de las comillas, también le indiquemos de qué color queremos que escriba el mensaje. Esto lo podemos realizar pulsando una combinación especial de teclas. Esas son:

CAPS SHIFT + SYMBOL SHIFT + Número del color. Con esta combinación de teclas cambiamos el color del papel. Primero pulsamos las teclas CAPS SHIFT y SYMBOL SHIFT a la vez, y cuando el cursor se ponga en modo E pulsamos la tecla que tiene el número que representa al color que queremos.

CAPS SHIFT + SYMBOL SHIFT CAPS SHIFT + Número de color. Con esta otra combinación podremos cambiar el color de la tinta. Primero pulsamos CAPS SHIFT y SYMBOL SHIFT a la vez y, cuando el cursor pase al modo E, pulsamos, también a la vez, las teclas CAPS SHIFT y el número del color que hemos elegido como tinta.

De esta manera no sólo ahorramos memoria, sino que además vemos en el listado cómo se van a imprimir las frases y mensajes.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

Con respecto al brillo, nos encontramos con que tenemos otra instrucción que nos permite cambiarlo. Esa instrucción es:

BRIGHT n. Donde «n» puede tener valor uno o cero. Si tenemos valor cero, el brillo será normal (o sin brillo). Si ponemos un uno (1), el brillo será fuerte (o con brillo).

El brillo afecta de igual manera a la tinta y al papel. Esto quiere decir que si un carácter de la pantalla tiene brillo uno, entonces el papel y la tinta de dicho carácter estarán brillantes. No se puede realizar de ninguna manera que mientras el papel de un carácter esté con brillo, la tinta esté sin él o al contrario.

También es posible introducir el comando BRIGHT en la sentencia PRINT. Hay dos formas de hacerlo. La primera de ellas es introduciendo dicha instrucción dentro del comando PRINT.

PRINT INK 3;BRIGHT 1;"HOLA"

La segunda es introduciendo el brillo después de las comillas del mensaje a imprimir. Para ello tenemos que pulsar la siguiente combinación de teclas.

CAPS SHIFT + SYMBOL SHIFT + 9. Conectamos el brillo a uno. Primero pulsamos las teclas CAPS SHIFT y SYMBOL SHIFT a la vez. Cuando el cursor se ponga en modo E pulsamos la tecla 9.

CAPS SHIFT + SYMBOL SHIFT + 8. Desconectamos el brillo. Primero pulsamos a la vez las teclas CAPS SHIFT y SYMBOL SHIFT. Cuando el cursor esté en modo E pulsamos la tecla 8.

Hay que decir que estas combinaciones de teclas no sirven para nada si no se encuentran después de las comillas del comando PRINT que controla el mensaje que queremos imprimir. Si las pulsamos entre la letras T de PRINT y las comillas, aunque veamos el mensaje tal y como queremos, éste no se imprimirá de esa manera, sino con los últimos valores directos de INK y de PAPER utilizados.

También hay que decir que los comandos de color, ya sea por instrucción o por la pulsación de una combinación de teclas, tienen prioridad sobre la última instrucción de

color dada. Dentro de estos dos tipos de introducción de color dentro de la sentencia PRINT, tiene más prioridad la combinación de teclas que la inclusión de comandos en la línea. Por ello, una línea como la siguiente:

INK 3:PRINT INK 4;"HOLA"

imprimirá la palabra HOLA en color 4 y no en color 3. Si dentro de las comillas hubiésemos pulsado la combinación de teclas que me dan el color 7, la palabra HOLA aparecerá en blanco.

Aunque no tiene mucho que ver con el color, el SPECTRUM nos permite imprimir mensajes parpadeantes. Esto lo podemos realizar con el comando FLASH. La acción de este comando es que, cuando se imprima un mensaje, primero lo imprimirá normal, pero un momento después lo imprimirá de forma que lo que tenía color de tinta ahora tiene el color del papel y lo que tenía color de papel pasa a ser del color de la tinta. Para entender mejor el funcionamiento de esta sentencia se recomienda introducir el programa número 4.

La sintaxis de esta sentencia es igual que la de BRIGHT.

FLASH 1. Hace que todo lo que se imprima a continuación aparezca parpadeando.

FLASH 0. Anula la acción de FLASH 1. Si no se hizo antes FLASH 1, el poner FLASH 0 no da ningún efecto.

El comando FLASH también puede ser incluido dentro de una línea de PRINT de las dos maneras que hemos visto. La combinación de teclas que nos permitirán hacer que podamos colocarlo dentro de las comillas de un PRINT son las siguientes.

CAPS SHIFT + SYMBOL SHIFT CAPS SHIFT + 9. Conecta el FLASH. Primero pulsamos las teclas CAPS SHIFT y SYMBOL SHIFT a la vez y cuando el cursor esté en modo E pulsamos las teclas CAPS SHIFT y 9 también a la vez.

CAPS SHIFT + SYMBOL SHIFT CAPS SHIFT + 8. Desconecta el FLASH. Primero tenemos que pulsar las teclas CAPS SHIFT y SYMBOL SHIFT a la vez. Cuando el cursor se ponga en modo E entonces pulsamos, también a la vez, las teclas CAPS SHIFT y 8.

```
100 REM *****
110 REM * PROGRAMA PARA VER EL USO *
120 REM * DEL COMANDO FLASH          *
130 REM *****
```

```

140 REM
150 PAPER 0
160 INK 6
170 BORDER 1
180 CLS
190 INPUT "DAME UN MENSAJE ";A$
200 PRINT A$
210 INPUT "EN QUE COLOR DE PAPEL LO IMPRIMO ";P
220 INPUT "CON QUE COLOR DE TINTA ";T
230 PRINT:PRINT:PRINT
240 PRINT "ESTOY IMPRIMIENDO EL MENSAJE ";FLASH 1;INK T;PAPER P;
A$;FLASH 0;PAPER 0;INK 6;"CON TINTA ";T;" Y PAPEL ";P

```

Lo último que nos queda por ver del SPECTRUM con respecto al color es la inversión del mismo. Puede haber veces que necesitemos imprimir mensajes en los que el papel tiene que tener el color de la tinta y la tinta el del papel. Para hacer esto no tenemos porqué tocar las sentencias de cambios de color. Para ello podemos hacer uso de la sentencia INVERSE. Como las dos últimas que hemos visto (BRIGHT y FLASH) el argumento que tenemos que poner detrás de INVERSE puede tener valor uno o valor 0. Si ponemos en uno (1) la inversión se hará efectiva. Si ponemos un cero, y antes habíamos realizado la inversión con INVERSE 1, el resultado será que a partir de ese momento todo lo que escribamos aparecerá de la forma normal.

Este comando puede encontrarse, al igual que todos los que hemos visto anteriormente, dentro y fuera de las sentencias PRINT. Para ponerlo dentro de una sentencia PRINT hay que seguir la misma regla que hemos visto anteriormente.

Si queremos realizar la inversión directamente sobre el texto que queremos imprimir, escribiendo el texto ya invertido, sólo tenemos que pulsar, después de las comillas del PRINT, las teclas CAPS SHIFT y 2 a la vez. Para anular la inversión basta con pulsar las teclas CAPS SHIFT y 3 a la vez.

El programa número 5 nos muestra un ejemplo de lo que acabamos de ver.

```

100 REM *****
110 REM * PROGRAMA PARA VER EL EFECTO *
120 REM * DEL COMANDO INVERSE EN EL *
130 REM * SPECTRUM. *
140 REM *****
150 REM
160 CLS
170 PRINT "ESTE TEXTO ESTA NORMAL"
180 INVERSE 1
190 PRINT "Y ESTE ESTA INVERTIDO"
200 INVERSE 0
210 INK 0
220 PAPER 6
230 PRINT "ESTE TEXTO ESTA NORMAL CON PAPEL AMARILLO Y
TINTA NEGRA"
240 INVERSE 1
250 PRINT "ESTE TEXTO ESTA INVERTIDO CON RESPECTO AL A
NTERIOR. NO SE HA CAMBIADO NI EL VALOR DEL PAPEL NI DE
LA TINTA"

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

Ya para terminar con el color en el SPECTRUM vamos a ver un programa que nos

realiza una demostración de todo lo que hemos visto.

```
100 REM *****
110 REM * PROGRAMA DEMOSTRACION *
120 REM * SOBRE TODOS LOS COMANDOS *
130 REM * DEL COLOR EN EL SPECTRUM *
140 REM *****
150 REM
160 CLS
180 DIM A$(8,8)
190 FOR I=1 TO 8
200   READ A$(I)
210 NEXT I
220 REM
230 DATA "NEGRO", "AZUL", "ROJO", "MAGENTA", "VERDE", "CIAN", "AMARI
LLO", "BLANCO"
240 REM
250 BRIGHT 0
260 LET B$="CERO"
270 FOR K=1 TO 2
280   FOR I=0 TO 7
290     FOR J=0 TO 7
300       INK J
310       PAPER P
320       BORDER J
330       PRINT "ESTE TEXTO ESTA CON TINTA ";A$(J+1);" Y PAPE
L ";A$(I); "CON BRILLO ";B$;" EL BORDE ES DE COLOR ";A$(J)
340     NEXT J
350   NEXT I
360   LET B$="UNO"
370   BRIGHT 1
380 NEXT K
390 PAPER 7
400 INK 1
410 BRIGHT 0
420 PRINT "ESTE TEXTO ESTA NORMAL Y CON BRILLO CERO"
430 FLASH 1
440 PRINT "ESTE TEXTO ESTA CON FLASH Y BRILLO CERO"
450 BRIGHT 1
460 FLASH 0
470 PRINT "ESTE TEXTO ESTA NORMAL Y CON BRILLO UNO"
480 FLASH 1
490 PRINT "ESTE TEXTO ESTA CON FLAS Y CON BRILLO UNO"
500 FLASH 0
510 INVERSE 1
520 PRINT "ESTE TEXTO ESTA INVERTIDO Y CON BRILLO UNO"
530 BRIGHT 0
540 PRINT "ESTE TEXTO ESTA INVERTIDO Y CON BRILLO CERO"
550 PAPER 7
560 INK 0
570 INVERSE 0
580 PAUSE 0
```

TRUCOS Y RUTINAS BASICAS



IGUIENDO con los programas de utilidad que nos permitirán definir nuestros propios caracteres y SPRITES sin tener que andar calculando y sin tener que utilizar ni papel ni lápiz, a continuación vamos a ver un genera-

dor de SPRITES para el MSX. Con este programa podremos definir SPRITES de 8×8 puntos o de 16×16 puntos. Podremos visualizarlos a tamaño normal o doble, podremos rotarlos, invertirlos, guardarlos en casete, leerlos de casete y algunas cosas más.

Al ejecutar el programa, éste nos preguntará qué tipo de SPRITE queremos definir. Se nos presentan dos opciones:

1. SPRITES de 8×8 puntos. Tamaño de un carácter.

2. SPRITES de 16×16 puntos. Como cuatro caracteres.

Una vez que hayamos pulsado la opción elegida nos aparecerá otro pequeño menú. Con el que podremos elegir la forma en la que queremos ver el SPRITE. Tenemos dos opciones:

1. SPRITES de tamaño natural (en 8×8 ó 16×16).

2. SPRITES ampliados al doble (en 8×8 ó 16×16).

Según pulsemos el 1 o el 2, así nos aparecerá por pantalla el SPRITE durante su definición.

```
1000 REM
1010 REM *****
1020 REM * GENERADOR DE SPRITES *
1030 REM *****
1040 REM
1050 SCREEN 0
1060 KEY OFF
1070 CLEAR 600
1080 DEF FN I$(C$,P,B)=LEFT$(C$,TP-1)+CHR$(B)+RIGHT$(C$,32-TP)
1090 DEF FN C(A,B)=B+16*INT(A/8)+1
1100 DEF FN M(A$,A,B)=ASC(MID$(C$,FNC(A,B)))
1110 REM
1120 REM *****
1130 REM * OPCION DE SPRITE *
1140 REM *****
1150 REM
1160 COLOR 11,0,0
1170 CLS
1180 LOCATE ,,0
1190 PRINT"  GENERADOR DE SPRITES"
1200 PRINT" ====="
1210 PRINT
1220 PRINT
1230 PRINT"Escoja tipo de SPRITE:"
1240 PRINT
```

TRUCOS Y RUTINAS BASICAS

```
1250 PRINT TAB(4);"1 - 8x8 Pixels."
1260 PRINT TAB(4);"2 - 16x16 Pixels."
1270 GOSUB 2270
1280 IF A$="1" THEN TA=7:L=TA ELSE TA=31:L=15
1290 PRINT
1300 PRINT "Escoja el tama$o del SPRITE:"
1310 PRINT
1320 PRINT TAB(4);"1 - Normal."
1330 PRINT TAB(4);"2 - Ampliado."
1340 GOSUB 2270
1350 PRINT
1360 PRINT
1370 PRINT " Utiliza las teclas del cursor paramoverte y la barra de espacio
paradibujar/borrar un punto."
1380 PRINT
1390 PRINT " Y PULSA UNA TECLA"
1400 A$=INKEY$
1410 IF A$="" THEN GOTO 1400
1420 SCREEN 1,VAL(A$)+1
1430 REM
1440 REM *****
1450 REM * VARIABLE DONDE ESTA *
1460 REM * EL SPRITE *
1470 REM *****
1480 REM
1490 FOR I=0 TO 31
1500 C$=C$+CHR$(0)
1510 NEXT I
1520 SPRITE$(0)=C$
1530 PUT SPRITE 0,(180,28),15
1540 SI$=CHR$(219)
1550 NO$=CHR$(196)
1560 CLS
1570 LOCATE 19,2
1580 PRINT"Sprite:"
1590 CX=0
1600 CY=0
1610 DIM C(31)
1620 GOSUB 1830
1630 FC=0
1640 ON STRIG GOSUB 2020
1650 STRIG(FC) ON
1660 ON KEY GOSUB 2720,2350,3070,2880,3650,3470,3830,4430
1670 FOR G=1 TO 8
1680 KEY(G) ON
1690 NEXT G
1700 GOSUB 4310
1710 REM
1720 REM *****
1730 REM * CONTROL DEL CURSOR *
1740 REM *****
1750 REM
1760 M=STICK(FC)
1770 IF M=0 THEN GOTO 1760
1780 CX=(CX-(M=3)+(M=7) AND L)
1790 CY=(CY-(M=5)+(M=1) AND L)
1800 GOSUB 2170
1810 GOTO 1760
1820 REM
1830 REM *****
1840 REM * IMPRESION DE LA PARRILA *
1850 REM *****
1860 REM
1870 LOCATE 0,0,0
```

```

1880 FOR Y=0 TO TA
1890   LOCATE 8*INT(Y/16),Y-16*INT(Y/16)
1900   T=128
1910   T1=ASC(MID$(C$,Y+1))
1920   FOR X=0 TO 7
1930     IF (T1 AND T)=T THEN PRINT SI$; ELSE PRINT NO$;
1940     T=T/2
1950   NEXT X
1960 NEXT Y
1970 SPRITE$(0)=C$
1980 BEEP
1990 GOSUB 2170
2000 RETURN
2010 REM
2020 REM *****
2030 REM * TRIGGER *
2040 REM *****
2050 REM
2060 TP=FNC(CX,CY)
2070 KO=FNM(C$,CX,CY)
2080 KO=(KO XOR 2^(7-(CX AND 7)))
2090 C$=FNI$(C$,TP,KO)
2100 GOSUB 2150
2110 SPRITE$(0)=C$
2120 RETURN
2130 REM
2140 REM *****
2150 REM * POSICIONADOR DEL CURSOR *
2160 REM *****
2170 REM
2180 FOR CC=1 TO 40
2190 NEXT CC
2200 LOCATE CX,CY,0
2210 PA=FNM(C$,CX,CY)
2220 IF (PA AND 2^(7-(CX AND 7))) THEN PRINT SI$; ELSE PRINT NO$;
2230 LOCATE CX,CY,1
2240 RETURN
2250 REM
2260 REM *****
2270 REM * ESPERA PULSACION *
2280 REM *****
2290 REM
2300 A$=INKEY$
2310 IF A$<>"1" AND A$<>"2" THEN 2300
2320 RETURN
2330 REM
2340 REM *****
2350 REM * ROTACION IZQUIERDA *
2360 REM *****
2370 REM
2380 GOSUB 2510
2390 IF TA=31 THEN 2440
2400 FOR I=0 TO 7
2410   C(I)=(C(I)*2 AND 254)+(C(I) AND 128)/128
2420 NEXT I
2430 GOTO 2620
2440 FOR I=0 TO 15
2450   T1=(C(I) AND 128)/128:T2=(C(I+16) AND 128)/128
2460   C(I)=(C(I)*2 AND 254)+T2:C(I+16)=(C(I+16)*2 AND 254)+T1
2470 NEXT I
2480 GOTO 2620
2490 REM
2500 REM *****
2510 REM * PASA LA MATRIZ C$ A *

```

TRUCOS Y RUTINAS BASICAS

```
2520 REM * LA MATRIZ C PARA EL *
2530 REM * CALCULO NUMERICO. *
2540 REM *****
2550 REM
2560 FOR I=0 TO 31
2570   C(I)=ASC(MID$(C$,I+1,1))
2580 NEXT I
2590 RETURN
2600 REM
2610 REM *****
2620 REM * SALIDA DE LAS RUTINAS *
2630 REM * DE ROTACIONES Y ESPEJOS *
2640 REM *****
2650 REM
2660 C$="":FOR I=0 TO 31:C$=C$+CHR$(C(I)):NEXT
2670 SPRITE$(0)=C$
2680 GOSUB 1830
2690 RETURN
2700 REM
2710 REM *****
2720 REM * ROTACION DERECHA *
2730 REM *****
2740 REM
2750 GOSUB 2510
2760 IF TA=31 THEN 2810
2770 FOR I=0 TO 7
2780   C(I)=C(I)/2+128*(C(I) AND 1)
2790 NEXT I
2800 GOTO 2620
2810 FOR I=0 TO 15
2820   T1=128*(C(I) AND 1):T2=128*(C(I+16) AND 1)
2830   C(I)=C(I)/2+T2:C(I+16)=C(I+16)/2+T1
2840 NEXT I
2850 GOTO 2620
2860 REM
2870 REM *****
2880 REM * ESPEJO VERTICAL *
2890 REM *****
2900 REM
2910 GOSUB 2510
2920 IF TA=31 THEN 2970
2930 FOR G=0 TO 3
2940   SWAP C(G),C(7-G)
2950 NEXT G
2960 GOTO 2620
2970 FOR G=0 TO 7
2980   SWAP C(G),C(15-G)
2990   SWAP C(G+16),C(31-G)
3000 NEXT G
3010 GOTO 2620
3020 REM
3030 REM *****
3040 REM * ESPEJO HORIZONTAL *
3050 REM *****
3060 REM
3070 GOSUB 2510
3080 IF TA=31 THEN 3180
3090 FOR G=0 TO 7
3100   T1=C(G)
3110   C(G)=0
3120   FOR I=0 TO 3
3130     C(G)=C(G)-(2^I)*((2^(7-I) AND T1)<>0)
3140     C(G)=C(G)-(2^(7-I))*((2^I AND T1)<>0)
3150   NEXT I
```

```

3160 NEXT G
3170 GOTO 2620
3180 FOR G=0 TO 15
3190   T1=C(G)
3200   T2=C(G+16)
3210   C(G)=0
3220   C(G+16)=0
3230   FOR I=0 TO 7
3240     C(G)=C(G)-(2^I)*((2^(7-I) AND T2)<>0)
3250     C(G+16)=C(G+16)-(2^(7-I))*((2^I AND T1)<>0)
3260   NEXT I
3270 NEXT G
3280 GOTO 2620
3290 REM
3300 REM *****
3310 REM * LOAD *
3320 REM *****
3330 REM
3340 GOSUB 3370
3350 REM
3360 REM *****
3370 REM * PREPARA INPUT *
3380 REM *****
3390 REM
3400 POKE &HF3F9,PEEK(&HF3FB)
3410 POKE &HF3F8,PEEK(&HF3FA)
3420 LOCATE 23,29
3430 PRINT STRING$(232,127)
3440 RETURN
3450 REM
3460 REM *****
3470 REM * SAVE *
3480 REM *****
3490 REM
3500 GOSUB 3370
3510 GOSUB 4140
3520 GOSUB 4230
3530 PRINT"      Y PULSE UNA TECLA";
3540 A$=INKEY$
3550 IF A$="" THEN 3540
3560 OPEN "cas:"+N$ FOR OUTPUT AS #1
3570 FOR G=0 TO TA
3580   PRINT#1,ASC(MID$(C$,G+1,1))
3590 NEXT
3600 CLOSE#1
3610 GOSUB 4310
3620 RETURN
3630 REM
3640 REM *****
3650 REM * LOAD *
3660 REM *****
3670 REM
3680 GOSUB 3370
3690 GOSUB 4140
3700 GOSUB 4230
3710 C$=""
3720 OPEN "cas:"+N$ FOR INPUT AS #1
3730 IF EOF(1) THEN 3770
3740 INPUT#1,A
3750 C$=C$+CHR$(A)
3760 GOTO 3730
3770 CLOSE#1
3780 GOSUB 1830
3790 GOSUB 4310

```

TRUCOS Y RUTINAS BASICAS

```
3800 RETURN
3810 REM
3820 REM *****
3830 REM * SAVE COMO LINEA DE DATA *
3840 REM *****
3850 REM
3860 GOSUB 3370
3870 GOSUB 4140
3880 INPUT "Numero de linea = ";LI
3890 GOSUB 4230
3900 PRINT"      Y PULSE UNA TECLA";
3910 A$=INKEY$
3920 IF A$="" THEN GOTO 3910
3930 OPEN "cas:"+N$ FOROUTPUT AS #1
3940 A$=STR$(LI)+" DATA "
3950 FOR G=1 TO TA
3960   A$=A$+"&H"+HEX$(ASC(MID$(C$,G,1)))+", "
3970 NEXT
3980 A$=A$+"&H"+HEX$(ASC(MID$(C$,TA+1,1)))
3990 PRINT#1,A$
4000 CLOSE#1
4010 GOSUB 3360
4020 LOCATE 0,17,0
4030 PRINT "Esta es la linea grabada:"
4040 PRINT
4050 PRINT A$
4060 PRINT
4070 PRINT "      Y PULSA UNA TECLA"
4080 A$=INKEY$
4090 IF A$="" THEN GOTO 4080
4100 GOSUB 4310
4110 RETURN
4120 REM
4130 REM *****
4140 REM * PREGUNTA NOMBRE *
4150 REM *****
4160 REM
4170 LOCATE 0,17
4180 INPUT "Nombre";N$
4190 IF N$="" THEN 4170
4200 RETURN
4210 REM
4220 REM *****
4230 REM * MENSAJES DE CASSETTE *
4240 REM *****
4250 REM
4260 LOCATE 0,22,0
4270 PRINT" Ponga en marcha el cassette"
4280 RETURN
4290 REM
4300 REM *****
4310 REM * PRINT MENU *
4320 REM *****
4330 REM
4340 LOCATE 29,23,0
4350 PRINT STRING$(232,127)
4360 LOCATE 0,16
4370 PRINT"Opciones:"
4380 PRINT"F1-Rota der.", "F5-Load", "F2-Rota izq.", "F6-Save norm.", "F3-Espejo hor
.F7-Save data.", "F4-Espejo ver.F8-Stop."
4390 GOSUB 2150
4400 RETURN
4410 REM
4420 REM *****
```

```

4430 REM * STOP *
4440 REM *****
4450 REM
4460 REM
4470 SPRITE$(0)=CHR$(0)
4480 CLS
4490 SCREEN 0
4500 PRINT " Si quieres definir otro SPRITE pulsa"
4510 PRINT "las teclas CTRL y STOP y haz RUN otra"
4520 PRINT "vez."
4530 GOTO 4530

```

A continuación nos aparece en la pantalla un mensaje que nos informa de la forma de mover el cursor. Para ello sólo tenemos que utilizar las teclas del cursor. Para iluminar un punto de la parrilla donde definiremos el SPRITE, sólo tenemos que pulsar la barra espaciadora. Para borrar un punto también tendremos que pulsarla. Podemos decir que la barra espaciadora se encarga de cambiar el estado del punto sobre el que se encuentra el cursor. Si éste está a 1 se pondrá a 0 y si está a 0 se pondrá a 1.

A continuación tendremos que pulsar una tecla para entrar de lleno en el programa. Una vez pulsada se borrará la pantalla y nos aparecerá lo siguiente:

— Una parrilla. En la esquina superior izquierda nos aparecerá una parrilla con 8×8 ó 16×16 puntos dependiendo del tipo de SPRITE que hayamos elegido al principio del programa. Dentro de esta parrilla dibujaremos el SPRITE hasta que esté completamente terminado.

— La palabra SPRITE en la parte superior derecha de la pantalla. Debajo de esta palabra nos irá apareciendo cómo va quedando el SPRITE según lo vamos definiendo. Su tamaño dependerá del tipo de SPRITE (8×8 ó 16×16) y de si queremos que aparezca con ampliación o sin ella.

— En la parte inferior de la pantalla nos aparecen los comandos de que disponemos y cómo llamarlos. Estos comandos se activan pulsando las teclas que van desde la F1 hasta la F8. El significado de cada comando es el siguiente:

— **F1.** Nos permite rotar el SPRITE un BIT hacia la derecha. El efecto que produce es como si se desplazase un punto hacia la derecha. Si alguna parte del SPRITE desaparece por la derecha volverá a aparecer por la izquierda.

— **F2.** Nos permite rotar el SPRITE un BIT hacia la izquierda. El efecto es el mismo que el de F1, pero el desplazamiento es en la dirección contraria.

— **F3.** Pulsando esta tecla el SPRITE toma la forma que tendría si se estuviese mirando a un espejo. El efecto resultante es que la primera línea del SPRITE pasa a ser la última, la segunda la penúltima, la tercera la antepenúltima y así sucesivamente, hasta que la última pasa a ser la primera. Es como darle la vuelta al SPRITE de arriba a abajo pasando por su eje horizontal.

— **F4.** Con esta tecla realizamos lo mismo que con F3, pero verticalmente. En este caso el eje de giro es vertical y lo que varían es la ordenación de las columnas. Así, si dibujamos un SPRITE con una cara que está mirando a la derecha, pulsando esta tecla, la cara mirará hacia la izquierda.

— **F5.** Con esta tecla cargamos un SPRITE que previamente hemos grabado. Como se puede apreciar hay dos opciones de grabación. Para que el programa pueda leer un SPRITE definido, desde el casete, tenemos que haberlo grabado usando la tecla F6. Ya veremos un poco más adelante el porqué.

— **F6.** Al pulsar esta tecla vamos a grabar el SPRITE de forma que si en otra ocasión queremos modificarlo, pueda ser leído desde el programa pulsando la tecla 5. Si grabamos el SPRITE pulsando esta tecla los datos que almacenamos no están disponibles por el usuario. Para que el usuario pueda utilizar el SPRITE que ha definido tiene que pulsar la tecla F7.

— **F7.** Esta opción nos graba en casete el SPRITE que acabamos de definir en forma de una línea DATA. De esta forma el usuario puede incluirla dentro de sus programas con sólo realizar un MERGE del fichero que crea el ordenador. El programa nos pide el número de línea que queremos que tenga dicha línea DATA y con dicho número de línea nos lo

TRUCOS Y RUTINAS BASICAS

grabará. También imprime dicha línea en la pantalla por si el usuario quiere copiarla a mano en vez de grabarla y luego hacer un MERGE. El formato en el cual graba esta línea el programa no puede ser leído más tarde usando la opción F5. Por ello se recomienda que, después de haber definido un SPRITE, el usuario lo grave de las dos maneras, pulsando F6 y F7.

— **F8.** Con la tecla F8 terminamos el programa y volvemos al BASIC.

Con el fin de no hacer el programa demasiado largo, éste se ha realizado de forma que sólo se pueda definir un SPRITE a la vez.

Si queremos definir más de un SPRITE, la forma de hacerlo es la siguiente:

1. Hacer RUN al programa.
2. Definir el SPRITE.
3. Grabar el SPRITE con la opción F5.
4. Grabar el SPRITE con la opción F6.
5. Pulsar la tecla F8.
6. Pulsar a la vez las teclas CONTROL y STOP.
7. Volver al punto número 1.

A continuación se proponen una serie de SPRITES en distintos tamaños para que el lector practique un poco.

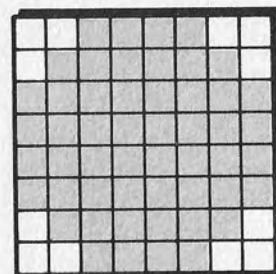
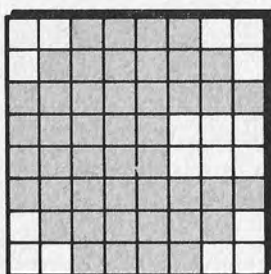
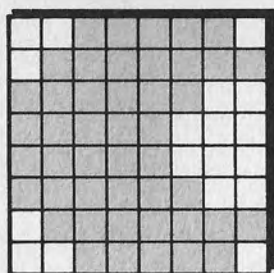


Fig. 1. Las tres posiciones distintas del Come-cocos.

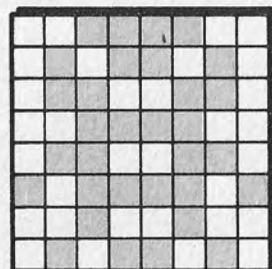
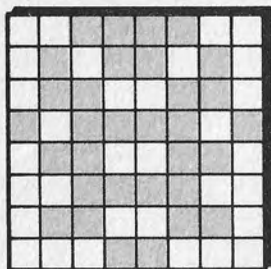


Fig. 2. El movimiento de una hormiga en dos caracteres (SPRITES).

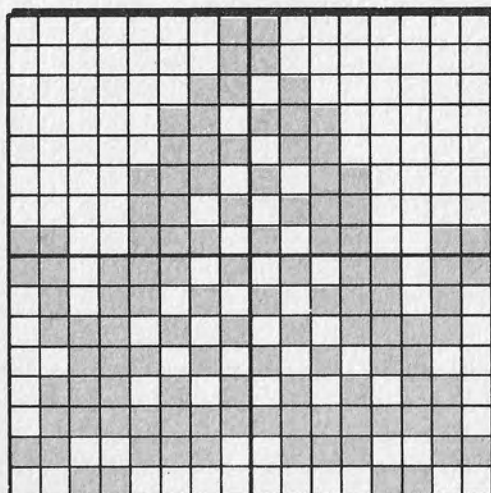


Fig. 3. Una nave espacial realizada con un SPRITE de 16 x 16.

■ Osciloscopio digital



Un instrumento que sin duda representa el sueño de todo aficionado al montaje de circuitos electrónicos es el osciloscopio y, más concretamente, el osciloscopio de memoria digital, que permita el almacenamiento de forma

permanente de las señales observadas.

Vamos a presentar un circuito, añadible como periférico a un ordenador personal de los que se están considerando como ejemplo en esta serie, que realiza las funciones de osciloscopio y que permitirá observar la mayoría de las señales presentes en nuestros proyectos. Por supuesto que presentará las limitaciones propias de los circuitos de adquisición económica en el mercado nacional, pero representará un salto importante en las posibilidades de nuestro taller, al poder ver las señales que hasta ahora intuíamos. En la pantalla de nuestro ordenador, acostumbrada a presentar marcianitos y monstruos espaciales, aparecerán las soñadas formas de onda de las se-

ñales que tantas veces hemos visto en los libros y revistas. Además de verlas podremos además «tratarlas» convenientemente con programas diseñados por nosotros mismos.

Las características que definen las prestaciones de un osciloscopio son: ancho de banda de la señal que pueden presentar, rango de amplitudes de la señal de entrada, número de canales, posibilidades de disparo y facilidades de la base de tiempos. Todas ellas son importantes en un equipo de uso continuado en el taller y su facilidad de manejo será la condición adicional para que se pueda realizar un trabajo cómodo y útil. En el equipo que podemos construir con nuestro OP haremos especial énfasis en la descripción de la función, para poder adaptar los componentes de que dispongamos a la realización propia.

Los bloques más importantes del osciloscopio son:

- Amplificador de entrada.
- Circuito de muestreo y retención.
- Conversor analógico/digital.
- Memoria de almacenamiento inter-

medio.

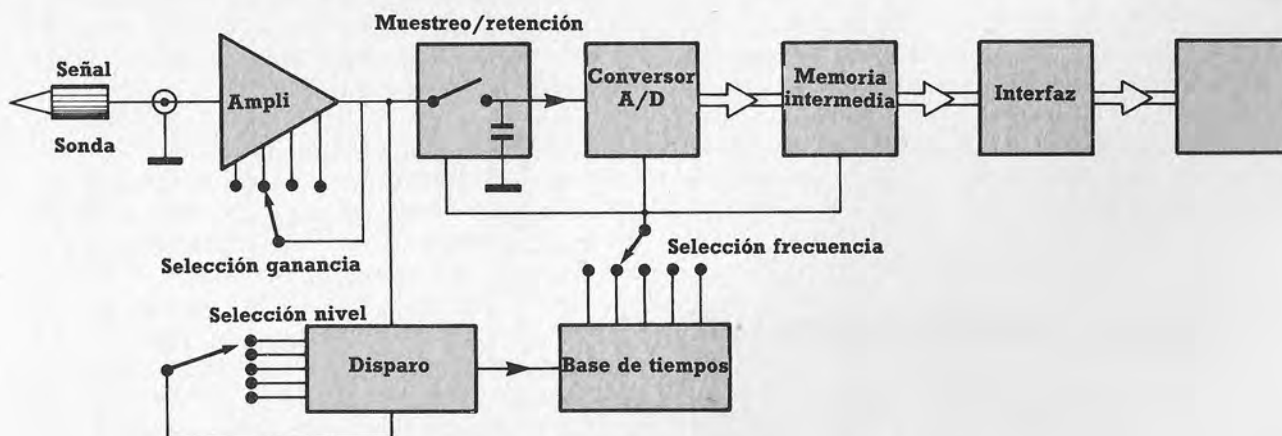


Fig. 1. Esquema de bloques de un osciloscopio digital.

Esquema de bloques

- Base de tiempos.
- Circuito de disparo.
- Interfaz con el ordenador.
- Panel de mandos.

El sistema en su conjunto es muy similar a los ya descritos para adquisición de datos analógicos y tratamiento digital de señales de audiofrecuencia. Presenta la diferencia fundamental en la velocidad de adquisición de datos y en la memoria intermedia de almacenamiento, que resulta imprescindible para poder almacenar los datos a la velocidad a que los suministra el conversor. Podría realizarse para alguno de los ordenadores personales mediante acceso directo a memoria

(DMA) pero la circuitería resultaría muy específica para cada caso. Con la solución de emplear una memoria externa se independizan prácticamente del ordenador. Las funciones de adquisición, solamente ligadas al programa mediante unos cuantos bits de control. Por otro lado el encarecimiento por usar una memoria externa resulta sobradamente compensado por la mejora de prestaciones.

Las funciones de presentación de la señal y control de los circuitos se harán de forma en parte programada y en parte manual.

Amplificador de entrada

Es un bloque muy importante de cara a la conservación de las características de la se-

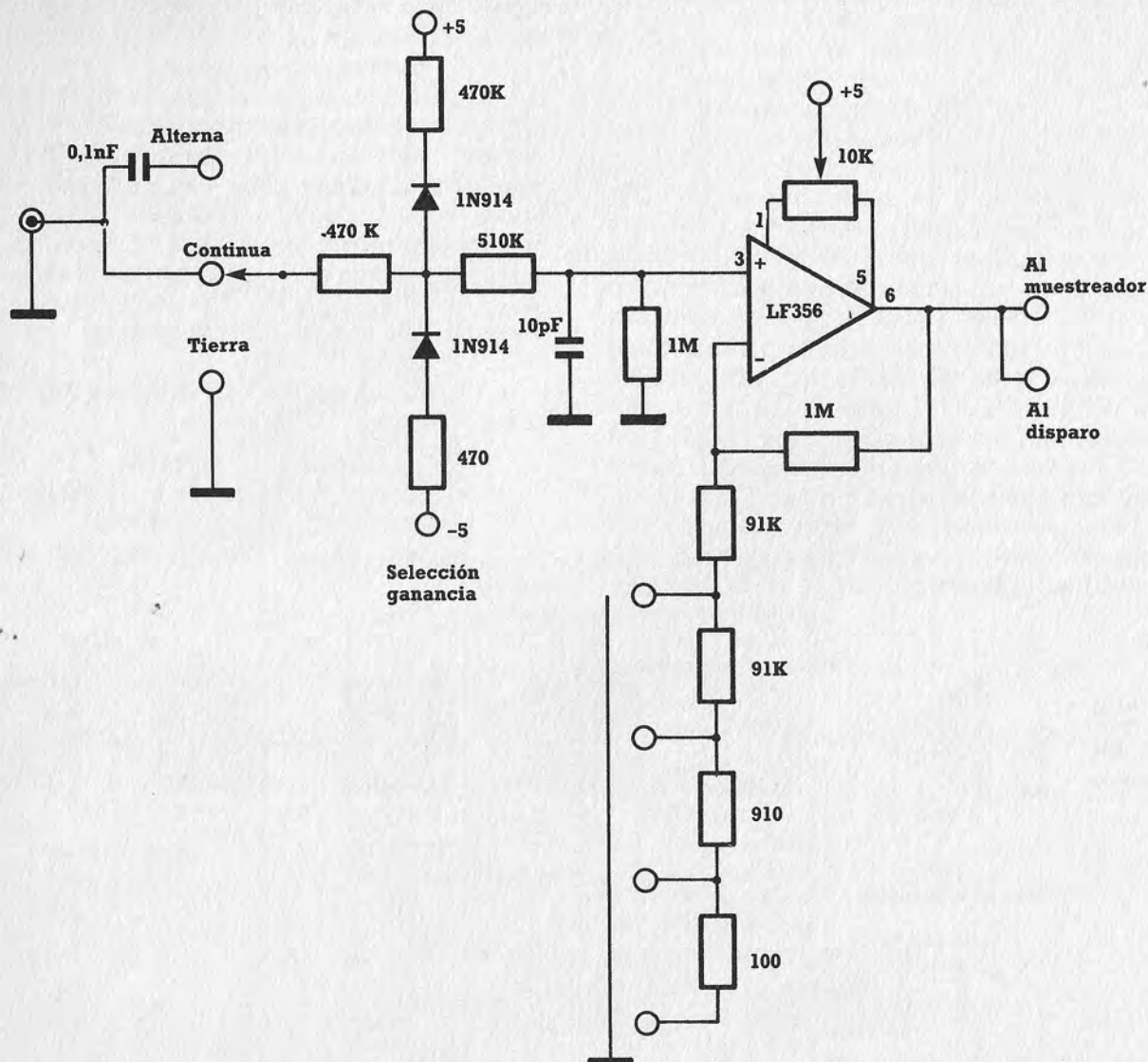


Fig. 2. Amplificador de entrada.

ñal de entrada. Debe adaptarse a los niveles y frecuencias de las señales a medir, causando la mínima influencia posible en el circuito que se ensaya. La impedancia de entrada del amplificador debe ser muy alta, para influir lo menos posible en el circuito. En los osciloscopios normales se obtiene el aumento de la impedancia de entrada mediante una sonda atenuadora, que si bien reduce la sensibilidad en un factor fijo, generalmente $\times 10$, reduce la influencia en los circuitos, sobre todo si son de alta impedancia. Además puede adaptarse el efecto de la capacidad de entrada mediante un pequeño condensador variable en paralelo con la resistencia de la sonda.

El circuito de entrada ha de estar muy protegido contra cualquier fallo en el circuito bajo ensayo y a cualquier manipulación incorrecta. Es necesario que no se estropee aunque estando en la escala de máxima sensibilidad le apliquemos la máxima tensión admisible. Por supuesto que si nos lo proponemos podemos lograr que hasta el mejor circuito de entrada falle, si le aplicamos señales fuera de los límites de su especificación. Como estamos diseñando un circuito para uso por nosotros mismos, deberíamos buscar una protección adaptada a nuestra posibilidad de equivocación. Se introduce la protección mediante diodos conectados a las fuentes de alimentación, después de la resistencia de entrada. Si la señal supera los valores de las tensiones positiva o negativa, se producirá una fuerte caída de tensión en la resistencia de entrada pues los diodos no permiten que suba por encima los niveles de recorte.

El amplificador de entrada ha de poseer ganancia variable, para poder llevar a los niveles apropiados del conversor cualquier señal que se nos presente. Se incluyen generalmente una parte con ganancia positiva y otra con atenuación, para poder observar señales de niveles más altos que los de los circuitos usuales. Realmente es mucho más fácil atenuar que amplificar.

El control de la ganancia del amplificador se realiza manualmente mediante un conmutador rotativo, que ajusta la ganancia en saltos, generalmente en la relación 1-2-5-10. Podría realizarse todo controlado por programa, montando todas las conmutaciones mediante relés, pero para un equipo de bajo coste no resulta recomendable. En cualquier caso es muy fácil sustituir el conmutador rotativo del esquema por relés, que realicen la misma función y que se conecten a sendos bits del

puerto de control. Se puede ajustar la ganancia mediante una resistencia variable. En su posición extrema deberá ser la de ganancia calibrada. Hay que poner por separado la resistencia de ajuste interna para la calibración de la de variación de la ganancia, con acceso desde el exterior.

En los osciloscopios de pantalla analógica se suele calibrar la ganancia en voltios por centímetro o por división, si es que la pantalla no está dividida en centímetros, para saber rápidamente el valor de la señal observada. En nuestro caso podremos calibrarlo según convenga, teniendo como principal preocupación el que la señal una vez amplificada no sobrepase los límites del conversor. Podemos indicar simplemente en las posiciones del conmutador rotativo el factor de ganancia correspondiente a cada una, para luego introducirlo en el menú que realiza la presentación en pantalla.

El circuito de entrada lleva la posibilidad de desplazar los niveles de las señales de alterna para hacerlos compatibles con el conversor, que sólo puede trabajar con señales de signo positivo. Esta función puede también hacerse accesible al panel de mandos, para poder situar la señal manualmente en el nivel más adecuado. Se calibrará también en el número de voltios desplazados sobre el valor de referencia.

Se suelen disponer como salidas accesibles en el panel de mandos varias señales de referencia, para calibrar el conjunto de sonda y amplificador antes de empezar a trabajar, si se va a medir con cierta precisión. Las señales de referencia pueden ser de continua o de alterna. Las señales de continua suelen ser un nivel patrón de 1 voltio. La señal de alterna suele ser una onda cuadrada de amplitud 1 voltio y flancos de subida muy rápidos, con lo que se ajusta tanto la ganancia como la respuesta en frecuencia de la sonda. Debe ajustarse el condensador variable hasta que la forma de la onda sea lo más cuadrada posible. Si además la frecuencia es generada con oscilador de cuarzo, sirve para calibrar el barrido.

■ Circuito de muestreo y retención

Para que la señal esté en un nivel estable al realizar la conversión a digital se incluye un circuito de muestreo y retención. La

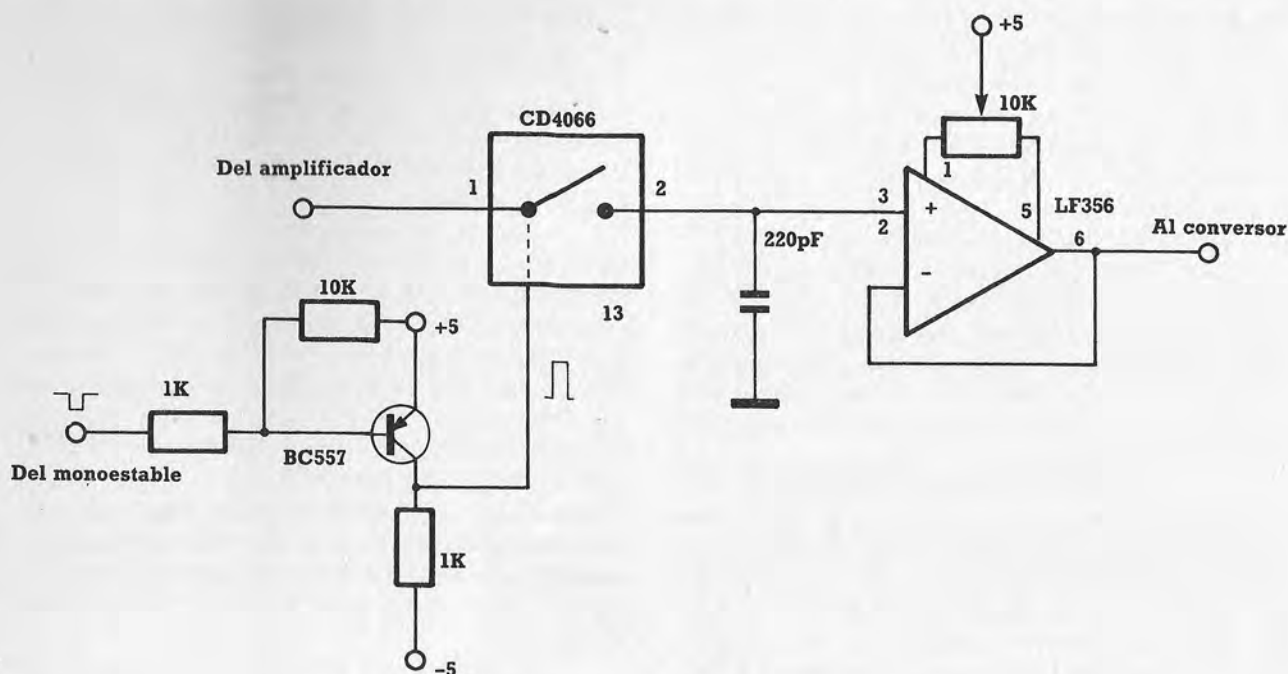


Fig. 3. Circuito de muestreo y retención.

apertura del muestreo se hace mucho antes del comienzo de la conversión, por el circuito de control. Un poco antes del inicio de la conversión se realiza el cierre de la muestra y el comienzo de la retención. La retención dura todo el período de conversión, pudiendo a continuación arrancar el comienzo de la nueva muestra.

El valor de la muestra se mantiene en un condensador, que alimenta un amplificador operacional de ganancia unidad, resultando así una salida de baja impedancia y la tensión proporcional al valor alcanzado en el momento del cierre del conmutador de entrada.

El conmutador está realizado con un circuito de puertas analógicas, que presenta muy baja resistencia en estado ON y muy alta impedancia en estado OFF.

■ Conversor analógico/digital

El circuito se realiza mediante dos circuitos integrados económicos: un conversor digital/analógico rápido de tipo DAC0801 y un registro de aproximaciones sucesivas, diseñado específicamente para esta función.

El conversor digital/analógico, que ya fue presentado en los primeros tomos sobre interfaces analógicas, realiza la conversión en modo corriente, de los 8 bits de la entrada, según los pesos indicados para cada una de ellas. Permite el ajuste de la escala de conversión, mediante el control de la corriente de referencia.

El registro de aproximaciones sucesivas realiza el proceso digital de la conversión. Presenta a las entradas del C D/A el bit de mayor peso y si el resultado de la comparación con la señal de entrada es positivo significa que la señal de entrada es superior a la de tanteo y por tanto hay que conservar el bit. En el período siguiente se tantea con el siguiente bit, dejándolo si el resultado es positivo y restaurándolo si el resultado es negativo. Así por aproximaciones sucesivas se obtiene al valor digital más parecido al de la señal de entrada, con un error menor que el correspondiente a la mitad del bit de menor peso. Al terminar la conversión el circuito da una señal indicadora de final, que iniciará el proceso de almacenamiento en la memoria.

El funcionamiento del circuito requiere un pulso de comienzo, pudiendo estar la entrada de reloj permanentemente activada. El pulso de comienzo se generará mediante una

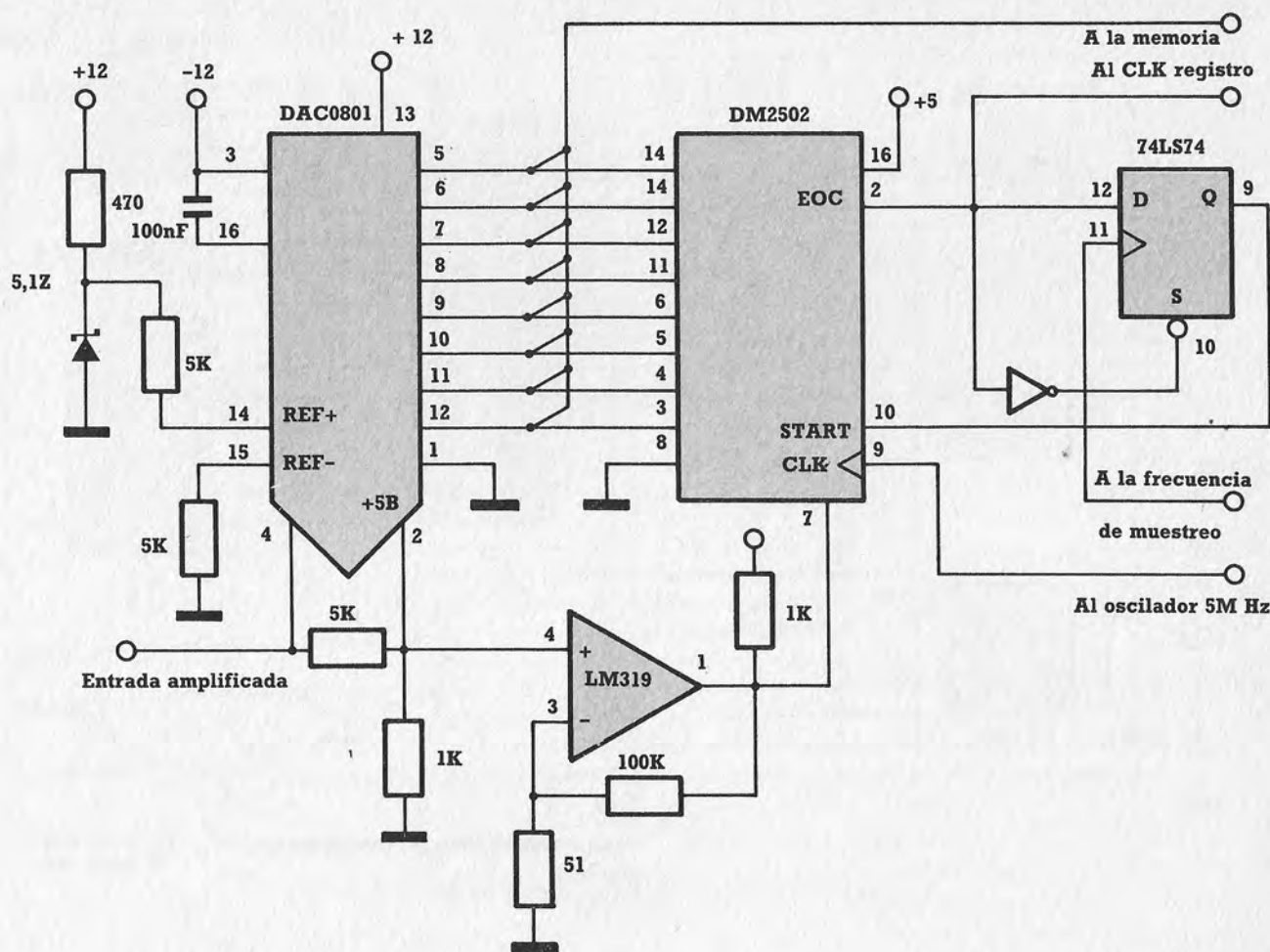


Fig. 4. Conversor analógico/digital.

señal derivada del generador de la base de tiempos.

El tiempo de conversión es de algo más de un microsegundo. Al final de la conversión en el registro de aproximaciones sucesivas está el último valor de prueba. Con el pulso de fin de conversión se almacena en el registro de mantenimiento y se da un pulso de escritura a la memoria intermedia. El pulso se genera mediante un monoestable.

Memoria intermedia de almacenamiento

Los valores medidos por el conversor están disponibles a la salida del registro de aproximaciones sucesivas en el momento de

producirse la señal de fin de conversión. Un registro formado por 8 biestables se carga coincidiendo con el pulso. Simultáneamente se inicia un ciclo de escritura en la memoria intermedia.

La memoria es de tipo estático, 4016 y contiene 2K octetos. Con el circuito indicado se encuentra permanentemente seleccionada. Se inicia un ciclo de escritura al activarse la entrada de -WE. Si esta entrada se encuentra en nivel alto estará disponible a las salidas internas el contenido de la posición seleccionada, de forma permanente, a partir del tiempo de acceso, es decir, no necesita pulso de lectura. Para poder leer el contenido en el exterior es necesario activar el control de salida -OE.

La utilización de la memoria es casi como de doble puerta, pues se escribe desde un circuito y se lee por otro. Para facilitar el

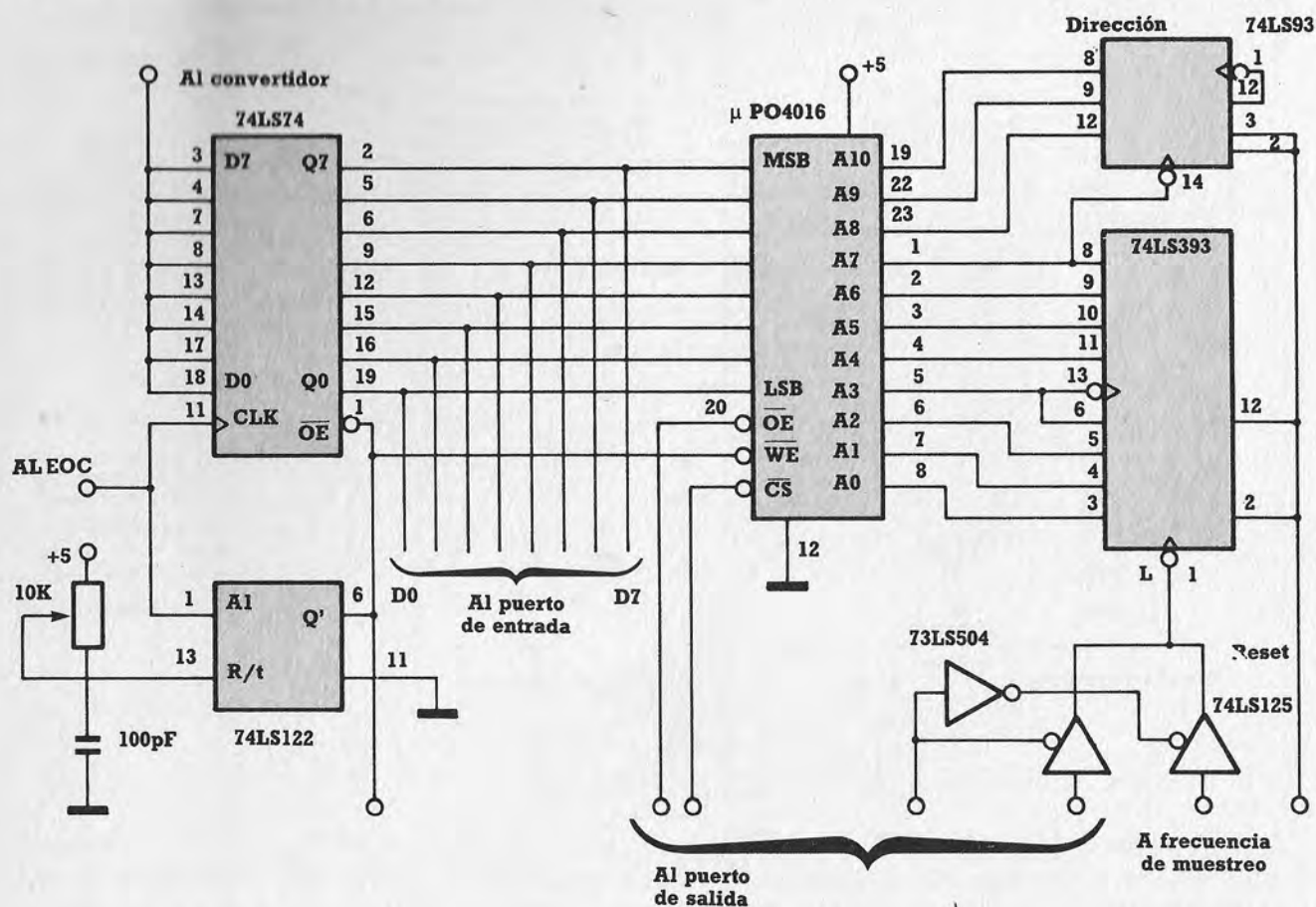


Fig. 5. Memoria intermedia.

direccionamiento en las dos fases de funcionamiento se incluye en el circuito un contador de direcciones que se emplea en ambos casos. La diferencia es que el avance de posición en un caso se hace mediante la señal de fin de conversión y en la otra a través de programa, para transferir el contenido de las medidas a la memoria principal.

El contador de direcciones se puede poner a cero mediante la activación de la señal de Borrado desde la interfaz. Normalmente se pondrá a cero el contador cada vez que se vaya a comenzar una medida o leer el contenido de la memoria.

Base de tiempos

La medida precisa de las señales requiere la toma de las muestras en instantes perfectamente definidos y repetibles. Se obtiene una secuencia de repetición de pulsos

de medida mediante un oscilador controlado por cuarzo y una serie de divisores. Un conmutador rotativo o un multiplexor permite seleccionar la frecuencia de repetición apropiada.

Para simplificar la generación de la frecuencia de muestreo se utilizan circuitos divisores por 10, que internamente se descomponen en divisor por 2 y divisor por 5. Con el esquema presentado se obtienen los intervalos entre muestras que se indican en cada posición. En la parte baja los saltos son 2-4-10, mientras que en la parte alta son 2-10. Pueden conseguirse otros períodos de muestreo, utilizando la misma técnica, pero deberá observarse que sean períodos iguales.

La base de tiempos estable se corresponde con la señal en diente de sierra utilizada en los osciloscopios ordinarios para hacer corresponder el desplazamiento en horizontal con el tiempo. En nuestro caso los incrementos de tiempo uniformes se consiguen con el oscilador estable y los divisores.

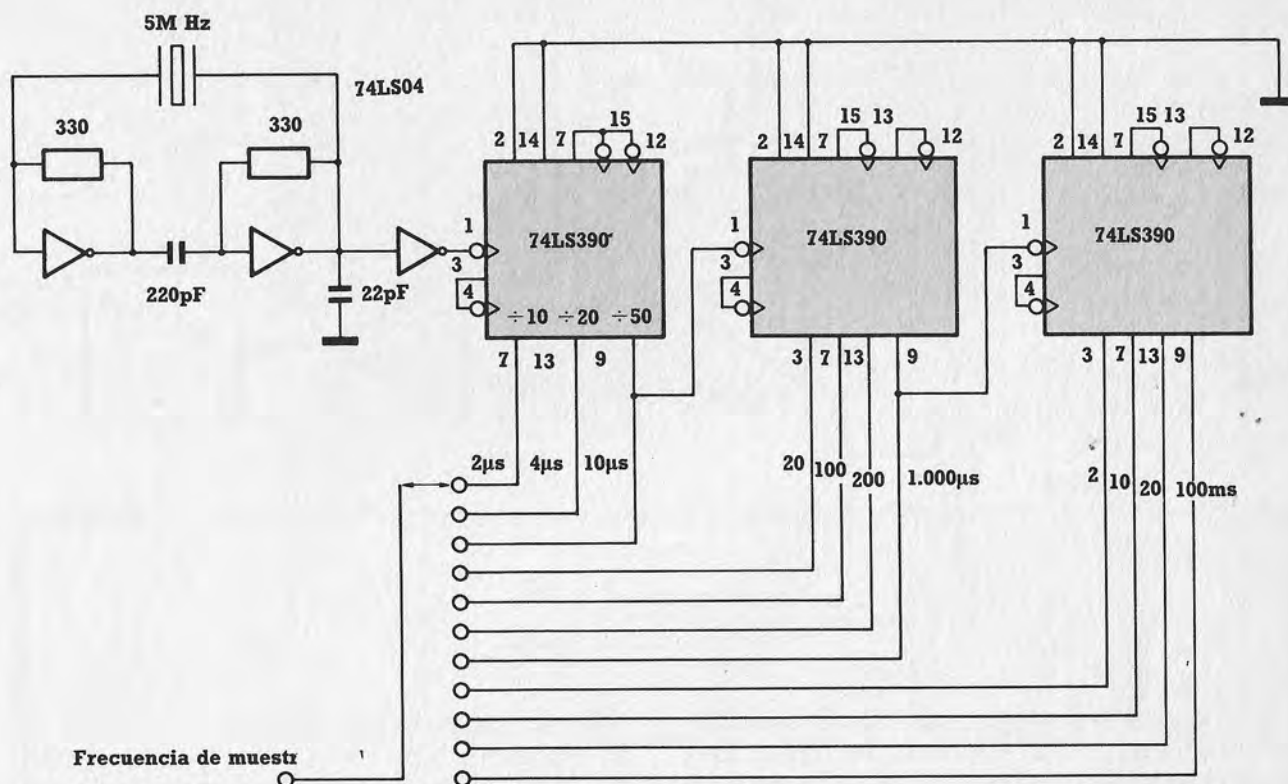


Fig. 6. Circuito generador de la base de tiempos.

El número de medidas efectuadas en cada barrido podríamos limitarlo por programa, llevando la cuenta del número de pulsos producidos. Lo más conveniente es efectuar un barrido completo cada vez, llenando la memoria intermedia y luego presentar en pantalla en sucesivas pasadas la información almacenada. El disponer de un número grande de muestras permite una gran flexibilidad para poder determinar problemas cuya relación causa-efecto está separada por un largo intervalo de tiempo.

El intervalo mínimo entre muestras, correspondiente a la frecuencia máxima de repetición de pulsos de 500 KHz, es de 2 microsegundos. Con esta velocidad de muestreo se pueden observar señales cuyo armónico superior sea inferior a 250 KHz. El ancho de banda por tanto es menor que el de la mayoría de los osciloscopios analógicos corrientes. Para aumentar el ancho de banda efectivo con señales repetitivas se recurre a un artificio muy interesante, consistente en disparar el barrido siempre en el mismo instante pero

realizar la toma de muestras con un ligero desplazamiento en cada pasada. Con esto se consigue componer la envolvente de la señal mediante puntos a lo largo del período de la señal.

Para señales de muy baja frecuencia el disparo del conversor podría realizarse por programa, sin necesidad de almacenar el valor leído en la memoria intermedia.

■ Circuito de disparo

El comienzo de la toma de muestras es de suma importancia en un circuito de medida. Según el problema a observar se presentan muy diferentes situaciones sobre cuál conviene que sea la condición que define el instante más oportuno de disparo para observar la señal un cierto tiempo antes y después del fenómeno que origina el disparo. Utilizando osciloscopio de tipo analógico es normal el ir tanteando diferentes frecuencias de barrido y

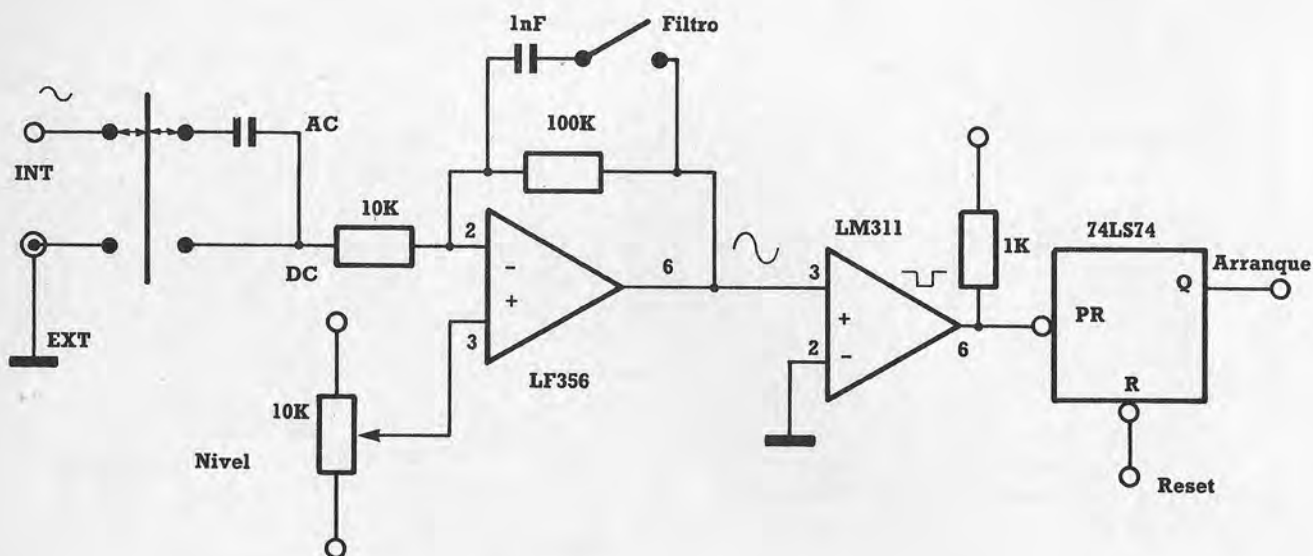


Fig. 7. Circuito de disparo.

condiciones de disparo hasta obtener una imagen estable. Con un osciloscopio digital este problema no se presenta así. La señal siempre se obtiene estable, pero puede que esté muy «lejos» en el tiempo del fenómeno que interesa observar. Es necesario condicionar mediante algún circuito adicional el comienzo del barrido y, por tanto, del almacenamiento de la señal.

Como condiciones mínimas de comienzo de barrido deberemos disponer de: nivel, subida o bajada y disparada desde el teclado, para fenómenos asociados a un suceso externo.

El problema es bien diferente en nuestro caso que en un osciloscopio ordinario, pues en éste debe producirse una señal repetitiva para que por persistencia de las imágenes en la pantalla se puede mantener la forma de onda. El osciloscopio digital resulta similar al trabajo en forma de un solo disparo, pues una vez producida la condición de inicio se efectúa la toma de datos y su almacenamiento en la memoria intermedia. No se necesita que la señal se repita para ser observada cómodamente. Sin embargo, deberemos poder identificar el fenómeno con alguna señal de referencia. Para poder hacer comparaciones en el osciloscopio ordinario de dos canales se realiza un muestreo de las señales observadas, para que haya correspondencia en el tiempo. Si las señales son repetitivas, parecerá que se producen simultáneamente, aunque en reali-

dad son presentadas de forma alternativa. En el osciloscopio de memoria la correspondencia deberá conseguirse por disparo del barrido con una señal de referencia externa igual en las diferentes circunstancias a comparar.

■ Interfaz de conexión al ordenador personal

Como en otros ejemplos anteriores conectaremos el nuevo equipo a través de la tarjeta de ampliación de puertos de entrada/salida. Se necesita un puerto de salida para las funciones de control y dos puertos de entrada para la lectura del contenido de la memoria externa y las señales de estado del circuito externo.

■ Montaje práctico

El osciloscopio debe ser una herramienta fácil y cómoda de usar por lo que debemos dedicar gran cuidado al diseño de la colocación de los mandos y las conexiones externas. Sobre una chapa de aluminio se colocarán los conmutadores rotativos, rotulando las posiciones con los valores que introducen en el sistema. Igualmente las resistencias variables de ajuste se actuarán mediante botones de man-

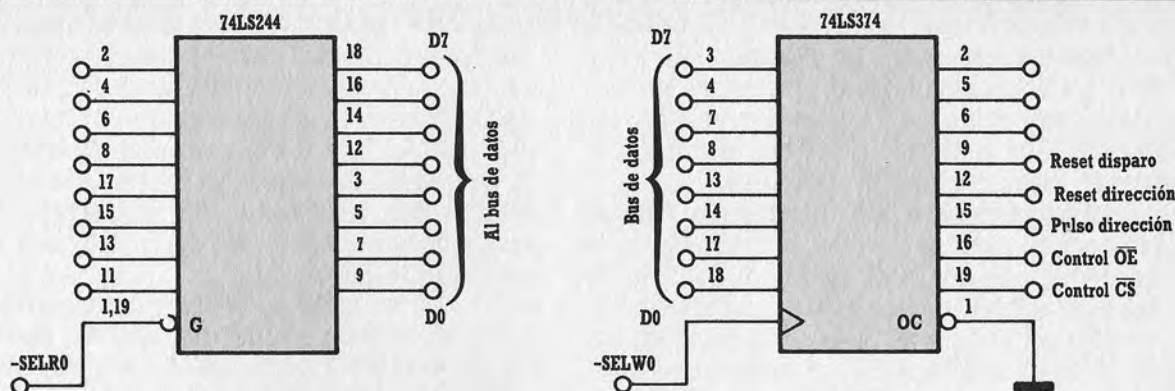


Fig. 8. Interfaz al ordenador personal.

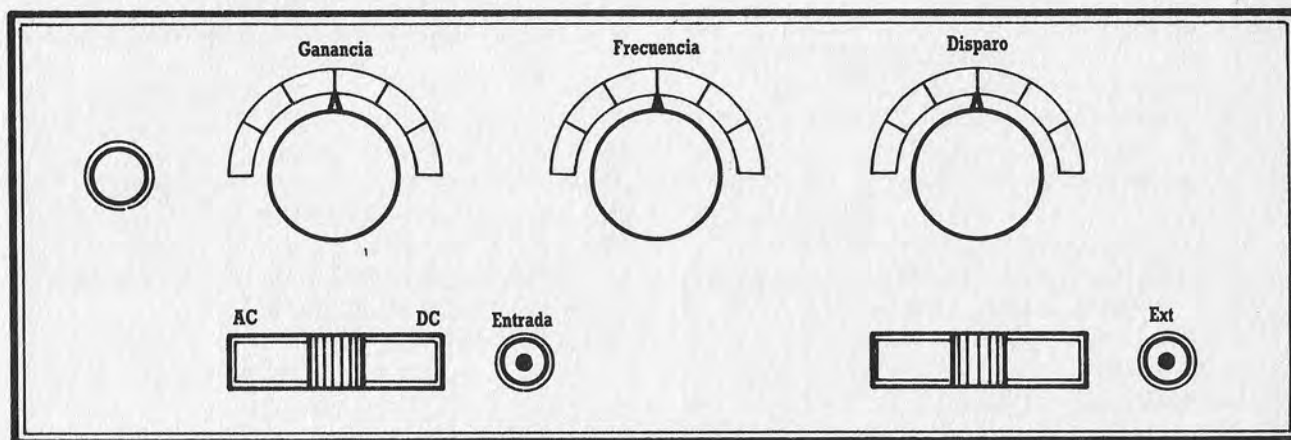


Fig. 9. Carátula del panel de mandos.

do con índice y dial de los valores que ajusta. Se marcarán especialmente las posiciones de calibración.

Las conexiones de las sondas y de las entradas de sincronización externa suelen hacerse mediante conectores BNC hembra. Para una aplicación de prueba hay conectores más económicos que pueden cumplir perfectamente, como los de tipo RCA empleados para las conexiones de audiofrecuencia. Es fundamental la correcta conexión de la masa de referencia. Si se adquieren sondas comerciales tendremos que poner los conectores BNC, pues prácticamente todas las que se venden presentan este tipo de conector.

Una precaución importante a tener en cuenta en los ensayos es la posibilidad de que la masa del sistema que vamos a medir presente tensión, o que incluso está conectada a uno de los hilos de la red de alimentación. Si éste es el caso deberemos tomar precau-

nes para realizar las medidas en flotante, como ya se indicó en el empleo de amplificadores operacionales de instrumentación.

Conclusión

El disponer de un ordenador personal detrás del sistema de adquisición de los datos permite muchas variaciones en la forma de utilización. Una vez adquirida la señal, puede presentarse de diferentes formas en la pantalla del ordenador: tal cual salió, amplificada, atenuada, filtrada, ampliada en el tiempo, para observación de fenómenos de corta duración, etcétera. Lo que resulta más interesante es que podemos presentarla transformada mediante operaciones programables y que resalten aspectos de interés. En el tomo anterior se mostró la posibilidad de conocer el contenido

frecuencial de una muestra, mediante la transformada rápida de Fourier. Podemos utilizar las funciones programadas para presentar los resultados de la transformación, junto con la forma de onda de la señal original.

Si disponemos de impresora resulta muy sencillo obtener una copia impresa de la señal capturada. Podemos utilizar la copia de la pantalla sobre la impresora o utilizando un programa un poco más elaborado explotar las posibilidades gráficas de la impresora para obtener una verdadera gráfica como si de un trazador de curvas se tratase.

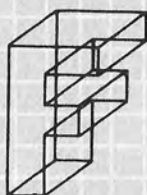
Otro aspecto no menos interesante es la posibilidad de comparar las señales recién to-

madas con otras almacenadas previamente. La comparación puede hacerse de forma manual, cotejando las representaciones gráficas de las dos señales, pero disponiendo de ordenador, la comparación se realiza mucho más cómodamente por programa. La diferencia puede cuantificarse y conocerse en qué instante se produce la diferencia.

En el tomo próximo presentaremos los programas de captura de señales, de control de la interfaz y de presentación de los resultados. Se podrá observar la potencia y eficacia de la interfaz diseñada y las posibilidades que tenemos de medida y análisis de las señales generadas en nuestro taller.

NATURALEZA Y TECNOLOGIA

■ Clasificación de los seres vivos



UE ya en el siglo XVII cuando se hizo la primera clasificación de seres vivos; y fue Linneo, un científico sueco, el primer hombre que hizo una clasificación de este tipo. También fue el que introdujo el concepto de especie, definiéndolo como aquel individuo que se adapta a las características de:

- Semejanza.
- Procedencia.

La ciencia que se encarga de estos estudios es la Taxonomía, que es la rama más antigua de las Ciencias Naturales y, aunque sea la que más lentamente evoluciona, es también la base y pilar de muchas otras ramas. Hoy en

día se siguen haciendo investigaciones relacionadas con esta ciencia.

Tanto así que en nuestros días hay diferenciaciones que no han podido determinarse muy claramente, y son ciencias como la Bioquímica y la Genética las que se encargan de estos estudios.

Durante mucho tiempo, la base de la clasificación de seres vivos fue la división que se hacía entre Reino Animal y Reino Vegetal; pero el descubrimiento de los microbios planteó nuevos problemas y así, a principios del siglo XX, Haeckel crea un nuevo reino llamado de los Protistas.

Pero a medida que la ciencia avanza se observa que este reino es insuficiente, y en los años 50, son Copeland y Whittaker quienes introducen un nuevo reino, el de las Moneras. Así pues, la clasificación actual queda dividida en cuatro reinos que son los que nos designa el programa en su menú principal, llamados Moneras, Protistas, Metafitas y Metazoos.

```

10 SCREEN 1
20 REM *****
30 REM ** CLASIFICACION SERES VIVOS **
40 REM *****
50 REM *****
60 REM ** MENU PRINCIPAL **
70 REM *****
80 CLS
90 GOSUB 2170
100 PRINT:PRINT
110 PRINT "
120 PRINT "
130 PRINT:PRINT
140 PRINT "
150 PRINT:PRINT
160 PRINT "
170 PRINT:PRINT
180 PRINT "
190 PRINT:PRINT
200 PRINT "
210 PRINT:PRINT
220 PRINT "
    
```

MENU	PRINCIPAL"
1 >	REINO MONERA"
2 >	REINO PROTISTA"
3 >	REINO METAZOO"
4 >	REINO METAFITA"
5 >	FIN DEL PROGRAMA"

```

230 LET X1=200:LET X2=80:LET D1=230:LET V1=210:GOSUB 2140
240 PRINT:PRINT
250 INPUT " OPCION ELEGIDA:";A
260 IF A=5 THEN END
270 IF A<1 OR A>4 THEN GOTO 80
280 ON A GOSUB 400,800,1200,1600
400 CLS:GOSUB 2170
410 PRINT:PRINT
420 PRINT "
430 PRINT "
440 PRINT:PRINT:PRINT
450 PRINT "
460 PRINT:PRINT:PRINT
470 PRINT "
480 LET X1=200:LET X2=80:LET D1=250:LET V1=140:GOSUB 2140
490 GOSUB 2210
500 GOTO 80
800 CLS:GOSUB 2170
810 PRINT:PRINT
820 PRINT "
830 PRINT "
840 PRINT:PRINT:PRINT
850 PRINT "
860 PRINT:PRINT:PRINT
870 PRINT "
880 PRINT:PRINT:PRINT
890 PRINT "
900 LET X1=200:LET X2=80:LET D1=300:LET V1=175:GOSUB 2140
910 GOSUB 2210
920 GOTO 80
1200 CLS:GOSUB 2170
1210 PRINT:PRINT
1220 PRINT "
1230 PRINT "
1240 PRINT:PRINT:PRINT
1250 PRINT "
1260 PRINT:PRINT:PRINT
1270 PRINT "
1280 LET X1=200:LET X2=80:LET D1=250:LET V1=140:GOSUB 2140
1290 GOSUB 2190
1300 IF B<>1 AND B<>2 AND B<>0 THEN GOTO 1200
1310 IF B=0 THEN GOTO 80
1320 IF B=1 THEN GOTO 1730
1330 GOTO 1920
1600 CLS:GOSUB 2170
1610 PRINT:PRINT
1620 PRINT "
1630 PRINT "
1640 PRINT:PRINT:PRINT
1650 PRINT "
1660 PRINT:PRINT:PRINT
1670 PRINT "
1680 PRINT:PRINT:PRINT
1690 PRINT "
1700 LET X1=200:LET X2=80:LET D1=225:LET V1=175:GOSUB 2140
1710 GOSUB 2210
1720 GOTO 80
1730 CLS:GOSUB 2170
1740 PRINT:PRINT
1750 PRINT "
1760 PRINT "
1770 PRINT:PRINT
1780 PRINT "
1790 PRINT "
1800 PRINT "
1810 PRINT "
1820 PRINT "
1830 PRINT "
1840 PRINT "
1850 PRINT "
1860 PRINT "
1870 PRINT "
1880 PRINT "
1890 PRINT "
1900 GOSUB 2210
1910 GOTO 1200
1920 CLS:GOSUB 2170
1930 PRINT:PRINT
1940 PRINT "
1950 PRINT "
1960 PRINT "

```

```

1970 PRINT "
1980 PRINT "
1990 PRINT "
2000 PRINT "
2010 PRINT "
2020 PRINT "
2030 PRINT "
2040 PRINT "
2050 PRINT "
2060 PRINT "
2070 PRINT "
2080 PRINT "
2090 PRINT "
2100 PRINT "
2110 PRINT "
2120 GOSUB 2210
2130 GOTO 1200
2140 PSET (X1,X2):LINE -STEP(D1,0)
2150 LINE -STEP(0,V1):LINE -STEP(-D1,0):LINE -STEP(0,-V1)
2160 RETURN
2170 PRINT " ** CLASIFICACION DE LOS SERES
      I V O S **"
2180 RETURN
2190 LOCATE 23,1:INPUT " OPCION ELEGIDA (0 PARA VOLVER AL MENU PRINCIPAL):";B
2200 RETURN
2210 LOCATE 23,1:PRINT "PULSE CUALQUIER TECLA PARA CONTINUAR"
2220 LET Z$=INKEY$:IF Z$="" THEN GOTO 2220
2230 RETURN

```

```

** INVERTEBRADOS **
-----"
ARTROPODOS"
  -Crustaceos"
  -Insectos"
  -Aracnidos"
  -Miriapodos"
CELENTEREOS"
ESPONGIARIOS"
ANELIDOS"
PLATELMINTOS"
NEMATELMINTOS"
MOLUSCOS"
EQUINOLERMOS"

```

Modificaciones para otros equipos

El programa es válido para IBM PC, aunque se puede ejecutar en cualquier ordenador de los que tratamos habitualmente, omitiendo la subrutina que recuadra los esquemas. De todas maneras existen algunas otras modificaciones que a continuación detallamos.

ZX-SPECTRUM:

Eliminar líneas 10, 230, 480, 900, 1280, 1700, 2140, 2150, 2160.

```

280 GOSUB 400*A
260 IF A=5 THEN GOTO 9999
2190 INPUT "OPCION ELEGIDA (0 PARA VOLVER AL MENU PRINCIPAL);B
2210 PRINT AT 21,0;"PULSE CUALQUIER TECLA PARA CONTINUAR"

```

COMMODORE:

Cambiar CLS por PRINT CHR\$(147) en líneas 80, 400, 800, 1200, 1600, 1730, 1920.

Eliminar líneas 10, 230, 480, 900, 1280, 1700, 2140, 2150, 2160.

Simular LOCATE con la subrutina gráfica de COMMODORE del tomo 1 de esta obra.

AMSTRAD:

Eliminar líneas 10, 230, 480, 900, 1280, 1700, 2140, 2150, 2160.

Sustituir LOCATE X,Y por LOCATE Y,X.

MSX:

Las mismas modificaciones que para AMSTRAD.

MATEMATICAS

Razones trigonométricas

La Trigonometría, como la Geometría y la Astronomía es una de las viejas herencias recibidas del mundo griego, cultivadas más tarde por los hindúes y los árabes que la fueron enriqueciendo con sus investigaciones.

El objeto de la Trigonometría está en buscar una relación entre los lados y los ángulos de un triángulo; y esta relación viene dada por las llamadas «RAZONES TRIGONOMETRICAS».

Con este programa se hace una representación gráfica de estas razones trigonométricas fundamentales, que son: seno, coseno y tangente de un cierto ángulo; junto con sus inversas; cosecante, secante y cotangente, respectivamente. Estas quedan representadas y definidas en el conjunto de ángulos de un plano formado en la llamada circunferencia goniométrica cuyo radio vale la unidad.

El concepto de estas razones trigonométricas ayuda a deducir a partir de ellas numerosas fórmulas, y a la vez numerosas apli-

APRENDER CON EL ORDENADOR

caciones que facilitan la resolución de diversos problemas ya no sólo en el ámbito de la Matemática, sino extendiéndose también al campo de la Física.

En el siguiente programa, en primer lugar, se nos pide el valor del ángulo en grados sexagesimales. Este ángulo no deberá exceder el valor de 360 grados, es decir, el recorrido completo a la circunferencia goniométrica.

Una vez introducido el ángulo irán apareciendo sucesivas pantallas en las que se representan las razones trigonométricas y su valor numérico. La representación de estas razones se hará mediante línea discontinua, en el caso de que el programa se ejecute en un ordenador IBM PC o compatible. En el resto de los ordenadores la razón trigonométrica aparecerá como una línea coloreada.

```

10 SCREEN 1
20 REM *****
30 REM ** RAZONES TRIGONOMETRICAS **
40 REM *****
50 PI=3.1415928#
60 REM *****
70 REM ** INSTRUCCIONES **
80 REM *****
90 CLS
100 PRINT "          T R I G O N O M E T R I A"
110 PRINT "          "
120 PRINT:PRINT:PRINT:PRINT
130 PRINT TAB(8);"*****"
140 PRINT TAB(8);"* ESTE PROGRAMA DIBUJA LAS RAZONES TRIGONOMETRICAS *"
150 PRINT TAB(8);"*
160 PRINT TAB(8);"*          DEL ANGULO QUE INTRODUZCAS          *"
170 PRINT TAB(8);"*****"
180 PRINT:PRINT:PRINT
190 PRINT TAB(18);"* PULSA UNA TECLA PARA COMENZAR *"
200 LET A$=INKEY$:IF A$="" THEN GOTO 200
210 CLS
220 INPUT "INTRODUZCA EL VALOR DEL ANGULO (en grados):";A
230 IF ABS(A)>360 THEN GOTO 210
240 REM *****
250 REM ** CALCULO DEL ANGULO EN RADIANES **
260 REM *****
270 LET B=A*2*PI/360
400 REM *****
410 REM ** DIBUJO DEL SENO **
420 REM *****
430 CLS
440 PRINT "          ** S E N O   D E   ";A;"   G R A D O S **"
450 LET X1=350:LET X2=151:LET R=67
460 GOSUB 2000
470 LINE (X1+R*COS(B),X2)-(X1+R*COS(B),X2-R*SIN(B)),,H9249
480 LOCATE 18,22:PRINT "SENO DE ";A;" GRADOS =" ;SIN(B)
490 GOSUB 4000
500 REM *****
510 REM ** DIBUJO DEL COSENO **
520 REM *****
530 CLS
540 PRINT "          ** C O S E N O   D E   ";A;"   G R A D O S *
540 PRINT "          "
550 LET X1=350:LET X2=151:LET R=67
560 GOSUB 2000
570 LINE (X1+R*COS(B),X2)-(X1+R*COS(B),X2-R*SIN(B))
580 LINE (X1+R*COS(B),X2-R*SIN(B))-(X1,X2-R*SIN(B)),,H9249
590 LOCATE 18,22:PRINT "COSENO DE ";A;" GRADOS =" ;COS(B)
600 GOSUB 4000
700 REM *****
710 REM ** DIBUJO DE LA TANGENTE **
720 REM *****
730 CLS
740 PRINT "          ** T A N G E N T E   D E   ";A;"   G R A D
0 S **"
750 IF ABS(A)=90 OR ABS(A)=270 THEN LET X1=350:LET X2=151:LET R=67:GOTO 820
760 GOSUB 2500
770 GOSUB 2000
780 LINE (X1,X2)-(X1+R,X2-R*TAN(B))
790 LINE (X1+R,X2)-(X1+R,X2-R*TAN(B)),,H9249
800 LOCATE 18,22:PRINT "TANGENTE DE ";A;" GRADOS =" ;TAN(B)
810 GOTO 890
820 GOSUB 2000
830 IF A=-90 OR A=270 THEN GOTO 870
840 LINE (X1+R,X2)-(X1+R,X2-67),,H9249
850 LOCATE 18,22:PRINT "TANGENTE DE ";A;" GRADOS =INFINITO"

```

```

860 GOTO 890
870 LINE (X1+R,X2)-(X1+R,X2+67),,,&H9249
880 LOCATE 18,22:PRINT "TANGENTE DE ";A;" GRADOS =-INFINITO"
890 GOSUB 4000
900 REM *****
910 REM ** DIBUJO DE LA COTANGENTE **
920 REM *****
930 CLS
940 PRINT "          ** C O T A N G E N T E   D E   ";A;"   G R A D
   O   S **"
950 IF A=0 OR ABS(A)=180 OR ABS(A)=360 THEN LET X1=350:LET X2=151:LET R=67:GOTO
1040
960 IF ABS(A)=270 OR ABS(A)=90 THEN LET C=0:GOTO 980
970 LET C=1/TAN(B)
980 GOSUB 3000
990 GOSUB 2000
1000 LINE (X1,X2)-(X1+R*C,X2-R)
1010 LINE (X1,X2-R)-(X1+R*C,X2-R),,,&H9249
1020 LOCATE 18,22:PRINT "COTANGENTE DE ";A;" GRADOS =" ;C
1030 GOTO 1110
1040 GOSUB 2000
1050 IF ABS(A)=180 THEN GOTO 1090
1060 LINE (X1,X2-R)-(X1+120,X2-R),,,&H9249
1070 LOCATE 18,22:PRINT "COTANGENTE DE ";A;" GRADOS =INFINITO"
1080 GOTO 1110
1090 LINE (X1,X2-R)-(X1-120,X2-R),,,&H9249
1100 LOCATE 18,22:PRINT "COTANGENTE DE ";A;" GRADOS =-INFINITO"
1110 GOSUB 4000
1200 REM *****
1210 REM ** DIBUJO DE LA SECANTE **
1220 REM *****
1230 CLS
1240 PRINT "          ** S E C A N T E   D E   ";A;"   G R A D O
   S **"
1250 IF ABS(A)=90 OR ABS(A)=270 THEN LET X1=350:LET X2=151:LET R=67:GOTO 1320
1260 GOSUB 2500
1270 GOSUB 2000
1280 LINE (X1+R,X2)-(X1+R,X2-R*TAN(B))
1290 LINE (X1,X2)-(X1+R,X2-R*TAN(B)),,,&H9249
1300 LOCATE 18,22:PRINT "SECANTE DE ";A;" GRADOS =" ;1/COS(B)
1310 GOTO 1390
1320 GOSUB 2000
1330 IF A=-90 OR A=270 THEN GOTO 1370
1340 LINE (X1,X2)-(X1,X2-80),,,&H9249
1350 LOCATE 18,22:PRINT "SECANTE DE ";A;" GRADOS =INFINITO"
1360 GOTO 1390
1370 LINE (X1,X2)-(X1,X2+80),,,&H9249
1380 LOCATE 18,22:PRINT "SECANTE DE ";A;" GRADOS =-INFINITO"
1390 GOSUB 4000
1400 REM *****
1410 REM ** DIBUJO DE LA COSECANTE **
1420 REM *****
1430 CLS
1440 PRINT "          ** C O S E C A N T E   D E   ";A;"   G R A D
   O   S **"
1450 IF A=0 OR ABS(A)=180 OR ABS(A)=360 THEN LET X1=350:LET X2=151:LET R=67:GOTO
1540
1460 IF ABS(A)=270 OR ABS(A)=90 THEN LET C=0:GOTO 1480
1470 LET C=1/TAN(B)
1480 GOSUB 3000
1490 GOSUB 2000
1500 LINE (X1,X2-R)-(X1+R*C,X2-R)
1510 LINE (X1,X2)-(X1+R*C,X2-R),,,&H9249
1520 LOCATE 18,22:PRINT "COSECANTE DE ";A;" GRADOS =" ;1/SIN(B)
1530 GOTO 5000
1540 GOSUB 2000
1550 IF ABS(A)=180 THEN GOTO 1590
1560 LINE (X1,X2)-(X1+120,X2),,,&H9249
1570 LOCATE 18,22:PRINT "COSECANTE DE ";A;" GRADOS =INFINITO"
1580 GOTO 5000
1590 LINE (X1,X2)-(X1-120,X2),,,&H9249
1600 LOCATE 18,22:PRINT "COSECANTE DE ";A;" GRADOS =-INFINITO"
1610 GOTO 5000
2000 REM *****
2010 REM ** DIBUJO DE LA CIRCUNFERENCIA **
2020 REM *****
2030 CIRCLE (X1,X2),R,,,1
2040 CIRCLE (X1,X2),R/5,,,0,B,1
2050 LINE (X1-R,X2)-(X1+R,X2)
2060 LINE (X1,X2+R)-(X1,X2-R)

```

APRENDER CON EL ORDENADOR

```

2070 LINE (X1,X2)-(X1+R*COS(B),X2-R*SIN(B))
2080 RETURN
2500 REM *****
2510 REM ** AJUSTE CIRCUNFERENCIA 1 **
2520 REM *****
2530 IF (A<-270 AND A>-360) OR (A<-90 AND A>-180) OR (A>0 AND A<90) OR (A<270 AND A>180) THEN LET I=1:GOTO 2550
2540 LET I=2
2550 LET R=135/(1+ABS(TAN(B)))
2560 IF R>67 THEN R=67
2570 LET X1=350
2580 IF I=1 THEN X2=159-R
2590 IF I=2 THEN X2=24+R
2600 RETURN
3000 REM *****
3010 REM ** AJUSTE CIRCUNFERENCIA 2 **
3020 REM *****
3030 IF (A<-270 AND A>-360) OR (A<-90 AND A>-180) OR (A>0 AND A<90) OR (A<270 AND A>180) THEN LET I=1:GOTO 3050
3040 LET I=2
3050 LET R=300/(1+ABS(1/TAN(B)))
3060 IF R>67 THEN R=67
3070 LET X2=151
3080 IF I=1 THEN X1=200+R
3090 IF I=2 THEN X1=400-R
3100 RETURN
4000 REM *****
4010 REM ** FINAL RAZON TRIGONOMETRICA **
4020 REM *****
4030 LOCATE 20,10:PRINT "* PULSE UNA TECLA PARA SIGUIENTE RAZON TRIGONOMETRICA *"
"
4040 LET Z$=INKEY$:IF Z$="" THEN GOTO 4040
4050 RETURN
5000 REM *****
5010 REM ** FINAL DEL PROGRAMA **
5020 REM *****
5030 LOCATE 20,17:PRINT "* PARA INTRODUCIR OTRO ANGULO PULSE S *"
5040 LET W$=INKEY$:IF W$="" THEN GOTO 5040
5050 IF W$="S" OR W$="s" THEN GOTO 210
5060 END

```

Modificaciones para otros equipos

El programa es válido para el ordenador IBM PC. Para el resto de las modificaciones son las siguientes:

ZX-SPECTRUM:

Suprimir línea 10.

Suprimir línea 50.

Sustituir todas las sentencias gráficas de la forma:

LINE (X,Y)-(X+A,Y+B)

por:

PLOT X,Y:DRAW A,-B

Sustituir todas las sentencias gráficas de la forma:

LINE (X,Y)-(X+A,Y+B),,,&H9249

por

PLOT X,Y:DRAW INK 2;A,-B

Sustituir las sentencias del tipo:

LOCATE X,Y:PRINT "..."

por:

PRINT AT X,Y;"..."

```

2030 CIRCLE X1,X2,R
2040 PLOT X1+R/5,X2:DRAW (X1+R*COS
      B)/5,(X2+R*SIN B)/5,B
2570 LET X1=128
2580 IF I=1 THEN LET X2=24+R
2590 IF I=2 THEN LET X2=159-R
3070 LET X2=91
3080 IF I=1 THEN LET X1=R
3090 IF I=2 THEN LET X1=255-R
5060 GOTO 9999

```

En las líneas 450, 550, 750, 950, 1250, 1450 las variables X1, X2 y R tendrán como valores:

X1 = 128 X2=91 R=67

COMMODORE:

Suprimir la línea 10.

Sustituir CLS por PRINT CHR\$(147) en líneas 90, 210, 430, 530, 730, 1230, 1430.

Simular los gráficos con subrutina gráfica especificada en el tomo 1 de esta obra.

Sustituir LOCATE X,Y:PRINT "..." por POKE 1024+Y+40*X: PRINT "..." en las líneas correspondientes.

AMSTRAD:

10 MODE 1

Sustituir LOCATE X,Y:PRINT "..." por LOCATE Y,X:PRINT "..." en líneas correspondientes.

Sustituir LINE (X,Y)-(X+A,Y+B) por PLOT X,Y:DRAWR A,-B en líneas correspondientes.

Sustituir LINE (X,Y)-(X+A,Y+B),,, &H9249 por PLOT X,Y:DRAWR A,-B,6 en líneas correspondientes.

Sustituir CLS por CLG en líneas 90, 210, 430, 530, 730, 930, 1230, 1430.

2030 ORIGIN X1,X2

2031 MOVE R,0

2032 FOR I=0 TO 2*PI STEP PI/180

2033 DRAW R*COS(I), R*SIN(I)

2034 NEXT I

2040 ORIGIN X1,X2

2041 MOVE R/5,0

2042 FOR I=0 TO B STEP PI/180

2043 DRAW R*COS(I)/5,R*SIN(I)/5

2044 NEXT I

SOCIEDAD

■ Industrias en España

Es siempre importante el nivel de industrialización de un país, por las connotaciones económicas y sociales que ello tiene a la hora de encuadrar ese país en un ámbito de industrialización; pero además de todo esto es también muy útil poder conocer la localización exacta de estas industrias, ya que esto nos permite darnos una idea de otras característi-

cas también muy importantes, como pueden ser la topografía de ese país, la riqueza del suelo, el tipo de ganado que existe, el desarrollo técnico, etc.

Por ello, en este programa mostramos un localización concreta de las industrias, dando una relación de las provincias en las que situaremos los más importantes complejos industriales.

De esta manera, podemos estudiar lo que genéricamente denominamos riqueza de una región o grado de industrialización de la misma. Y observamos cómo hay provincias que, o bien por su situación geográfica o bien por otros factores, como pueden ser incluso la historia que haya tenido esa región, son más ricas que otras que se encuentran en condiciones más desfavorables. Y cómo se localizan industrias similares en zonas próximas y de cultivos semejantes.

Otro factor que se puede deducir de la localización industrial son los diferentes climas que reinan en esa zona.

El programa que a continuación se presenta nos ofrece una panorámica general de la situación industrial española. Introduciendo el nombre de la provincia de la que se desea conocer su industria, el ordenador nos dará una relación de las principales industrias de ese lugar.

Hay que tener en cuenta que se debe introducir el nombre completo de la provincia. Por ejemplo, el caso de Castellón de la Plana. También cabe resaltar que se ha omitido en lo posible el nombre de las comunidades autónomas, a excepción de la provincia de La Rioja, que ha sido introducida en el programa con este nombre.

```
10 REM *****
20 REM ***** INDUSTRIAS EN ESPAÑA *****
30 REM *****
40 DIM I$(34):DIM P$(47):DIM N(47,30)
50 CLS
60 LOCATE 9,13:PRINT "*****"
70 LOCATE 10,13:PRINT "** I N D U S T R I A S **"
80 LOCATE 12,13:PRINT "** E N E S P A Ñ A **"
90 LOCATE 13,13:PRINT "*****"
100 FOR I=1 TO 34
110 READ I$(I)
120 NEXT I
130 FOR I=1 TO 47
140 READ P$(I)
150 NEXT I
160 FOR I=1 TO 47
170 FOR J=1 TO 30
180 READ N(I,J)
190 NEXT J
200 NEXT I
210 CLS:LET C=0
220 PRINT " I N D U S T R I A S E N E S P A Ñ
```

46


```

10 REM *****
20 REM * COMIENZO DEL PROGRAMA *
30 REM *****
40 CLS
50 LET C=0
60 RANDOMIZE TIMER
70 LET N=INT (RND*10000)+1
80 LET A=N
90 LOCATE 10,20:PRINT "POR QUE NUMERO SE PUEDE DIVIDIR ";A;"?"
100 LOCATE 15,27:INPUT X:IF X=0 THEN GOTO 100
110 IF X=1 THEN LOCATE 17,22:PRINT "ESO YA LO SABIA YO":FOR I=1 TO 1000:NEXT I:C
LS:GOTO 90
120 IF X=A THEN GOTO 230
130 IF ((A/X)-INT(A/X))=0 THEN LOCATE 20,20:PRINT ";; C O R R E C T O !!":GOTO
180
140 LOCATE 20,27:PRINT ";; N O !!":LET C=C+1
150 FOR I=1 TO 1000:NEXT I
160 CLS
170 GOTO 90
180 FOR I=1 TO 1000:NEXT I
190 CLS
200 PRINT A;" / ";X;" = ";A/X
210 LET A=A/X
220 GOTO 90
230 CLS
240 LOCATE 10,25:PRINT "LA DIVISION HA TERMINADO."
250 LOCATE 12,25:PRINT "HAS TENIDO ";C;" FALLOS"
260 LOCATE 14,10:PRINT N;" = ";:GOSUB 1040
270 LOCATE 16,20:PRINT "SI QUIERES PROBAR OTRA VEZ PULSA ESPACIO"
280 LET Z$=INKEY$:IF Z$=" " THEN GOTO 280
290 IF Z$=" " THEN GOTO 40
300 END
1000 REM *****
1010 REM ** DESCOMPOSICION EN FACTORES PRIMOS **
1020 REM ** DEL NUMERO ELEGIDO ALEATORIAMENTE **
1030 REM *****
1040 GOTO 1080
1050 PRINT " 2 x ";
1060 LET N=N/2
1070 IF N-1=0 THEN GOTO 1190
1080 IF ((N/2)-INT(N/2))=0 THEN GOTO 1050
1090 LET M=3
1100 LET Q=SQR(N)+1
1110 IF M>=Q THEN GOTO 1180
1120 IF ((N/M)-INT(N/M))=0 THEN GOTO 1140
1130 LET M=M+2:GOTO 1110
1140 IF N/M*M-N=0 THEN GOTO 1160
1150 GOTO 1130
1160 PRINT M;" x ";
1170 LET N=N/M:GOTO 1100
1180 PRINT N;" x ";
1190 PRINT " 1":RETURN

```

Modificaciones para otros equipos

El programa es válido para el IBM PC. Las modificaciones para los demás equipos son:

ZX-SPECTRUM:

60 RANDOMIZE 0

Sustituir LOCATE X,Y:PRINT "..." por PRINT AT X,Y;"..." en las sentencias correspondientes.

300 GOTO 9999

AMSTRAD:

60 RANDOMIZE TIME

Sustituir LOCATE X,Y por LOCATE Y,X en las sentencias adecuadas.

COMMODORE:

40 PRINT CHR\$(147)

Suprimir línea 60.

Para simular LOCATE X,Y:PRINT "..." utilizar la rutina del tomo 1 para situar el cursor en una posición determinada de la pantalla.

160 PRINT CHR\$(147)

190 PRINT CHR\$(147)

Cambiar CLS por PRINT CHR\$(147) en línea 110.

230 PRINT CHR\$(147)

MSX:

Las mismas modificaciones que para AMSTRAD.

PEQUEÑA HISTORIA DE LA INFORMATICA

■ Los comienzos de una gran multinacional, la IBM



OMO bien sabemos, los orígenes de la IBM están en la fusión de tres empresas, que se unieron en 1911 para formar la Computing Tabulating Recording Company. Una de las empresas fundadoras fue la compañía de ta-

buladoras de Herman Hollerith, y las otras dos se dedicaban a maquinaria específica de precisión, aunque no tenían nada que ver con las tareas de oficina; una de ellas fabricaba balanzas de precisión, y la otra, relojes. Dichas empresas tampoco estaban creadas y habían ido creciendo con el tiempo. Eran producto de la unión de otras muchas empresas bastante más pequeñas, que se iban fusionando para disponer de mayores recursos.

En los años siguientes a su creación, la CTR no era una empresa demasiado rentable. Apenas si obtenía beneficios, y no conseguía vencer el mercado potencial al que tenía que enfrentarse.

Con el fin de cambiar la situación, se nombró director general a un hombre que, al menos, había demostrado ser un magnífico trabajador y un excelente vendedor: Thomas Watson. Al año de la gestión de Watson, la empresa había crecido y contaba con casi mil quinientos empleados, obteniendo unos beneficios realmente altísimos, casi el 30 %.

El secreto de su éxito fue una política de seriedad, eficacia, agresividad. Además, reinvertía para poder expandirse con firmeza. Poco tiempo después, hacia el final de la Pri-

mera Guerra Mundial, Watson había conseguido que las ventas se cuadruplicaran, aunque los beneficios no llegaron al esperado 30 %, meta que Watson se había propuesto, y que intentaba que todos sus empleados le ayudaran a conseguir.

La compañía matriz, CTR, tenía contactos en el extranjero, principalmente en Alemania e Inglaterra. Watson cuidó mucho de mantener esos contactos y clientes, mejorando sus relaciones sustancialmente.

Como también conocerá el lector, por haberlo mencionado en algún otro momento, en 1924, Watson decide cambiar el nombre a la compañía, y le impone el que había seleccionado en 1917 para su sucursal canadiense, International Business Machines. En cierto modo, este pequeño dato que parece no tener importancia, supone un cambio de rumbo. En ese momento, la dirección de la compañía se decanta por dejar de lado las balanzas y relojes, y centrarse, fundamentalmente, en la maquinaria de tarjetas perforadas, típica de resolución de tareas de oficina. (La sucursal de IBM en Canadá suponía una fábrica de fichas y una planta de montaje.)

Watson siguió con su política de expansión, creando nuevas plantas de montaje y fábricas de tarjetas en puntos clave del continente europeo. Así, en 1922, lo hace en París, donde establece la Société Internationale des Machines Commerciales, que se dedicaba primordialmente a tabuladoras tipo Hollerith.

En el año 1929 se produce la caída de la Bolsa de Nueva York. Sin embargo, la compañía subsiste, y consigue pagar dividendos a duras penas. A pesar de todo, sus ingresos brutos superan los quince millones de dólares

PEQUEÑA HISTORIA DE LA INFORMATICA

por esas fechas, y su personal ha crecido muchísimo.

Los años de depresión afectaron a la compañía, pero no todo lo que se piensa. En primer lugar, no todo el campo industrial se vio afectado del mismo modo, y en segundo lugar, el Gobierno hacía enormes esfuerzos para canalizar los recursos del modo más eficaz. Todo este tipo de tareas exigía máquinas como las que podía ofrecer la IBM, de forma que en esos años el cliente principal de la gran compañía fue el Gobierno americano.

La IBM ofrecía dos sistemas principales para la adquisición de sus máquinas: la compra, o el alquiler. Como las máquinas eran muy caras, muchas empresas no contaban con recursos suficientes para comprarlas, con lo que se decidían por alquilarlas. De este modo, además de rebajar el precio mensual, consideraban que obtendrían un mejor servicio por parte de la empresa fabricante, como, por otro lado, era cierto. (Sin embargo, la primacía del servicio al cliente fue siempre una norma de la empresa, que mimaba a sus empleados, a los que exigía una total dedicación.) Así pues, los ingresos de la compañía, en aquellos años difíciles de depresión eran, a pesar de todo, suficientes, gracias a los alquileres y también a las fichas perforadas que utilizaban las máquinas, y que debían ser compradas exclusivamente a la casa fabricante, ya que, en caso contrario, era causa de rescisión de contrato.

Pero no todos los ingresos de la compañía provenían del continente americano. La IBM se había expandido muchísimo, y ya estaba establecida en más de sesenta países, en los cuales, apenas si fabricaba nada, enviando el material desde los Estados Unidos para, a lo sumo, ser montado en el país que lo importaba.

Con la política de reinversión de recursos, la empresa disponía de capital suficiente para ir renovando y adaptando su tecnología. Sin embargo, en este momento, Thomas Watson ya había envejecido, y era mucho más conservador. En 1925, apareció un nuevo modelo de clasificadora horizontal; unos años después se modificó el diseño de las fichas, pasando a tener ochenta columnas, pero no hubo muchos cambios hasta 1931, en que la empresa presentó la serie 600, que eran unas máquinas que realizaban multiplicaciones y divisiones. Realmente no puede considerarse que evolucionara a ritmo poderoso, más bien parecía un caracol, incapaz de salir de su cómoda concha, conocida y confortable.

Sin embargo, la empresa tampoco ofre-

¿Sabía usted que...

Vamos a contar al lector una pequeña anécdota sobre la IBM, que tiene dos versiones diferentes.

Estando dos ejecutivos de una empresa tomando una cola en una pequeña terraza de la ciudad donde la Gran Multinacional tiene su sede central, vieron pasar un enorme cortejo de elegantes coches negros. Uno de los ejecutivos dijo al otro: «Qué cortejo fúnebre más impresionante.» «No es un cortejo fúnebre —contestó el otro—, se trata, simplemente, de los abogados de IBM, que vuelven del restaurante.»

La segunda versión, simplemente constata que se trataba de los abogados de IBM, ocupados en algún juicio importante, que volvían después del descanso de la comida.

cía grandes beneficios a los accionistas. Watson estaba llevando a cabo una política expansionista en el extranjero, de la que obtendría excelentes resultados a medio y largo plazo, pero que no resultaba muy clara para el accionista medio.

En los años siguientes, la compañía cambió algo su rumbo, haciéndose menos conservadora. En 1933, adquirió otra compañía de máquinas de escribir, dato que parece muy sensato, dado que el mercado de IBM para sus otras máquinas eran precisamente las oficinas, en las que la máquina se había hecho imprescindible. Sin embargo, poco tiempo después, la IBM sacó un nuevo modelo de máquina de escribir, eléctrica. Ya empezaba a derivar recursos para la investigación.

A finales de los años treinta, la compañía ya era un monstruo enorme, con más de diez mil empleados y unos ingresos brutos de más de treinta millones de dólares. Los beneficios no habían vuelto a alcanzar las expectativas del famoso 30 % si bien es cierto que estaban muy cerca de alcanzarlo.

La Segunda Guerra Mundial no supuso desastre alguno para la IBM. Es bien cierto que fue indemnizada por pérdidas en los países vencidos, pero la verdad es que creció enormemente, superando los ciento cincuenta millones de dólares de facturación al final de la contienda.

Además, en tiempo de guerra es normal que las compañías interesantes, desde algún punto de vista, se reconviertan para fabri-

car material de guerra. Así les había ocurrido a innumerables compañías de máquinas de escribir, y otro tipo de máquinas especializadas. Pero para esos momentos, la IBM tenía muchos contactos en Washington y su potencia era increíble. De forma que consiguió convencer a la Administración de que el material que fabricaba era muy interesante para la industria de guerra, evitando la reconversión. La realidad es que así era, en cierto modo, aunque también podían ser útiles otras compañías que habían sido militarizadas. Además, la empresa estaba colaborando con cierto éxito en labores de criptografía, consideradas de importancia fundamental para el Gobierno americano.

Para los años cincuenta, la compañía estaba establecida en casi cualquier país del mundo. Quizá el mercado anglosajón fue de los últimos en los que consiguió penetrar, ya que tenía contactos con una compañía inglesa, la BTC (British Tabulation Machines Company), que actuaba como subsidiaria de la IBM.

En los años que siguieron, y tras algunos esfuerzos para readaptarse y comenzar a fabricar ordenadores, la empresa creció hasta alcanzar los 600 millones de dólares, cifra increíble. Su personal había crecido casi al mismo ritmo, pasando a ser más de 70.000 personas. La compañía había multiplicado sus ingresos por catorce en dieciocho años. Todo un récord.

Durante los años cincuenta, sesenta y al comienzo de los setenta, la compañía se enfrentaría a importantes marcas de la competencia, algunas realmente no tan importantes. Pero antes de todo lo anterior, y que relataremos en otro momento con más detalle, naturalmente, la compañía pasó a fabricar ordenadores. Dicen (las malas lenguas, desde luego) que el hecho que impulsó a IBM a pasar a la fabricación de ordenadores se basaba en la sensación de desprestigio por haber perdido

la cuenta del censo. Nosotros no lo creemos así. La verdad, es que esa cuenta es simplemente un valor sentimental, sin demasiado interés, ya que el Gobierno de los Estados Unidos había demostrado, de mil formas diferentes, su consideración por la compañía. En 1930, la Oficina del Censo disponía de maquinaria propia, pero también tenía máquinas IBM.

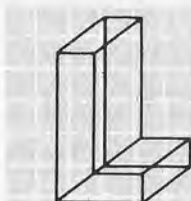
Además de las consideraciones anteriores, la verdad es que junto a Watson se encontraba un nuevo equipo de personas mucho más jóvenes, entre los que descollaba su hijo mayor, Thomas Watson junior, que abogaban por un cambio de rumbo de la empresa, para pasar a fabricar otro tipo de máquinas universales, mucho más complejas, que requerían una gran inversión en investigación, pero que hasta el momento eran «asequibles» para la gran empresa. Es probable que si la IBM no se hubiera decidido en aquel momento hacia el cambio, no hubiera obtenido la posición que ha ido alcanzando en los años sucesivos.

Tom Watson junior había sido educado por su padre para sucederle, igual que los reyes preparan a sus hijos, con todo cuidado. Fue a las mejores universidades (hablaremos de ello más adelante) y continuó su educación en Europa. Pero quizá una de las cosas que más marcaron a Watson junior fue su participación como piloto en la Segunda Guerra Mundial. Este hecho no parece tener mucha relación con su cargo en la compañía que dirigía su padre, pero sí lo tuvo, ya que se rodeó de un equipo de veteranos de guerra, gente experimentada, y también moderna, que veía los problemas desde una perspectiva más real que la de Watson senior, que, a pesar de su enorme valía y sentido común, estaba dando muestras de un conservadurismo excesivo.

Seguiremos hablando de la gran multinacional, del primer ordenador, de la serie 360 y de otras muchas cosas que quedan por decir en estas cortas líneas

SIMULADORES

■ Simuladores de navegación



OS simuladores son equipos utilísimos. A pesar de su precio, con frecuencia muy alto, ahorran mucho dinero.

Los pilotos y navegantes son personas que con frecuencia manejan equipos importantes y valiosos. Por tanto, tienen una enorme responsabilidad entre sus manos. Esa responsabilidad y la experiencia son cualidades que se van desarrollando con los años, y las vivencias. En los tiempos actuales, la experiencia debe obtenerse rápidamente, y no se puede despilfarrar tiempo ni errores para adquirirla.

El navegante utiliza el simulador para adquirir experiencia. Navega por aguas conocidas y también desconocidas. Se le presentan situaciones nuevas, que debe resolver reaccionando con habilidad. Desde su puesto puede observar la ventana del puente, y verá que ésta va cambiando por todos los agentes que actúan. Agentes externos, como las condiciones meteorológicas, posiciones y reacciones de otros barcos, y producidas por el propio piloto, que maneja los mandos del barco. Puede navegar de día y de noche, acercarse a un puerto o sortear escollos, observar señales desde otros barcos o las estrellas que brillan en el firmamento, y que debe reconocer. Todo se comporta del mismo modo que si estuviera en el mar, enfrentándose al destino.

Los simuladores son útiles para muchos objetivos distintos, veamos algunos:

a) Sirven para iniciar a los alumnos en tareas de pilotaje. De este modo, se evitan los errores por falta de experiencia, cuyas causas pueden deteriorar equipos, frecuentemente muy caros, e incluso el propio barco.

b) Sirven para entrenamiento y puesta

a punto para oficiales con experiencia, por ejemplo, para pasar de un determinado barco a otro más grande, o distinto en algún aspecto.

c) Preparan a las personas a enfrentarse a situaciones peligrosas o difíciles, de forma que en el momento en que se produzcan ya estén familiarizados con ellas, y no pierdan un tiempo en pensar posibles salidas, tiempo que puede ser importante en resolver la situación.

d) Se gana en seguridad, por todas las razones anteriormente expuestas.

e) Se estudian efectos especiales al acercarse a puertos, a barcos, etc.

f) Sirven para investigar en el diseño de barcos, puertos, etc.

Para crear un simulador, lo primero que se hace es medir todas las características del barco para, posteriormente, realizar los modelos matemáticos en un ordenador de simulación. El modelo debe actuar a los controles del puente de mando, y responder adecuadamente. La posición y el rumbo son datos que se van suministrando continuamente desde el ordenador de simulación al ordenador que crea los escenarios, que, a su vez, mantiene un modelo matemático completo de toda la escena que aparezca dentro de una determinada distancia, con un horizonte específico. Además, el ordenador escenario actualiza los valores de posición y rumbos de los «otros barcos» y objetos móviles de la escena. En cuanto a dichos barcos y otros objetos cambiantes de la escena, de los que hemos hablado, también pueden programarse o ser controlados exteriormente por un monitor. El cuadro completo puede entonces enviarse al generador de escenarios, que traduce dicho dibujo en señales adecuadas para ser presentadas en el monitor de vídeo. Para que el proceso tenga un realismo aceptable, es necesario que todas las tareas anteriores se realicen unas veinticinco veces por segundo. En la figura que

mostramos, las señales de vídeo se envían a cinco proyectores que proyectan imágenes sobre la parte posterior de una pantalla de visualización. Dicha pantalla se encuentra delante del puente de mando, siendo independiente de los proyectores. El campo de visión que se ve desde el puente está formado por la aportación de cada proyector, 40 grados. Por tanto, en nuestro caso, será de 200 grados.

Las escenas que maneja el ordenador de escenarios se mantienen como un conjunto de modelos, cada uno de los cuales está definido por una serie de imágenes planas. En el momento de introducirlas dentro del campo visual, las imágenes se procesan rápidamente, para que sólo las que sean visibles al espectador pasen al generador de imágenes. Una escena contiene muchos elementos típicos, como, por ejemplo, los barcos de que hablamos anteriormente, puertos, olas, islas, montañas, olas producidas por otras embarcaciones, etc. Todos estos elementos combinados, producen una simulación real, vívida y muy útil.

■ Simuladores de vuelo

Echemos un rápido vistazo de en qué consiste un simulador de vuelo. Uno de los más utilizados es el CT-5, angloamericano.

El uso inmediato de los simuladores de vuelo es el entrenamiento de pilotos de aviación. Los primeros simuladores aparecieron en los años sesenta, y han ido mejorando mucho con el tiempo, siguiendo fielmente el desarrollo espectacular de la industria informática.

Las escenas que aparecen en los últimos modelos son de un realismo extraordinario, y además disponen de una amplísima gama de modelos de aviones. El número de simuladores que existen en el mercado también ha aumentado mucho.

Los simuladores imitan virtualmente muchísimos tipos de avión, reproducen aeropuertos, pistas de aterrizaje, *oilrigs* en el mar, y otros entornos posibles.

Los principales compradores de este tipo de equipos son las grandes compañías de aviación, el Ejército del Aire, y algunas pequeñas compañías, dedicadas a la preparación de pilotos. Apenas si existen simuladores para pilotaje de helicópteros.

Para cada uno de los casos, el simulador debe disponer del cuadro de mandos

completo de cada aparato, aunque bien es cierto que muchos aparatos disponen de cuadros de mando muy similares. El piloto se sienta en la cabina de control, y puede ver por la ventana los mismos elementos que vería si estuviera sentado en su avión. La ventana dispone de sistemas sofisticadísimos que reproducen exactamente lo que ocurre en el exterior y el comportamiento del avión con las acciones del piloto. Existen, incluso, algunos simuladores que reproducen la acción del viento sobre la cabina, inclinándola en función, tanto del ángulo de incidencia del viento sobre ella, como de las velocidades respectivas, viento, nave. La pantalla debe ser capaz de mostrar detalles difíciles de interpretar en imágenes, como viento, arena, niebla, etc.

La estructura del sistema puede verse en la figura. El piloto se encuentra sentado en la cabina, unos proyectores que se encuentran detrás y sobre él proyectan imágenes sobre la pantalla.

Los simuladores son útiles para resolver algunos tipos de problemas; veamos algunos de ellos:

Sirven para entrenar a la tripulación en la resolución de situaciones extremas o peligrosas, como fuego en el aparato, mal funcionamiento de controles o mandos, malas condiciones climatológicas, colisiones, aterrizajes de emergencia, etc. Es fundamental que se dote de experiencia válida a las tripulaciones ante tales circunstancias, y todo ello sin peligro para el piloto. El ahorro en material es además considerable. Este extremo es a tener en cuenta, ya que el coste por hora de pilotaje es altísimo y, aunque los buenos simuladores son caros, el gasto se reduce muchísimo (supone aproximadamente un 10 por 100 del precio de entrenamiento con un aparato). Además, aprovechan los mejores instructores, que no se cansan, viajando constantemente, y pueden ser mucho más útiles en tierra, donde se producen esas «terribles condiciones», que afortunadamente no se dan en vuelo todos los días.

El desarrollo de un nuevo simulador requiere una inversión considerable. En primer lugar, obliga a la creación de modelos matemáticos que reproduzcan las condiciones para pasar, a continuación, al desarrollo del software, que reproduzca tales condiciones de entorno. También deben duplicarse e introducirse en el simulador, los controles físicos. El trabajo completo puede durar más de dos o tres años.

Las imágenes que se suministran al sistema de presentación difieren de un sistema a otro. Sin embargo, el realismo depende del número de polígonos (o superficies) que puedan procesarse para formar la imagen que está en continua transformación, y el tipo de sistema utilizado para presentar la imagen. Naturalmente, a mayor número de polígonos que se puedan generar, más detalles de todo tipo tendrá la imagen, y más sensación de realismo. Estos polígonos (o superficies) se toman de una base de datos muy grande. El número de detalles también se incrementa al acercarse a superficies concretas, donde es evidente que tienen mucha mayor importancia detalles como el tipo de terreno, etc. Entre las posibles escenas, deben encontrarse aeropuertos concretos, con sus peculiaridades específicas, pero también, como antes hemos comentado, condiciones externas atípicas, vientos, pistas de aterrizaje cortas o faltas de pista de aterrizaje, etc.

Hemos de indicar también que los simuladores tienen una importancia enorme en investigación. Suponen un medio importante para comprobar las reacciones del avión, ante distintos diseños, de sus partes, proporcionando un medio rápido y barato de acercamiento a la solución idónea.

También sirven para probar los distintos tipos de avión, y estudiar sus características, fallos, etc., comparativamente, forzando condiciones extraordinarias sin riesgo ni coste.

Después de estudiar de forma muy general las cualidades de los simuladores, veamos también someramente su estructura, compuesta por un procesador de propósito general (que maneja los datos, procesa los modelos matemáticos, modifica los datos con los parámetros variables que se le introduzcan, etcétera), un procesador de imágenes (que prepara los datos para su presentación visual) y un elemento de proyección de las imágenes (pantalla de representación de los escenarios). A su vez, el procesador de imágenes (hardware específico y clave del simulador) está compuesto por cuatro elementos fundamentales (en el orden en que intervienen para crear el escenario): el manejador de objetos, el manejador de polígonos, el procesador geométrico y el procesador de presentación.

La tarea fundamental del manejador de objetos es, partiendo del punto de vista del piloto, producir elementos para la escena. Para ello debe dirigirse a la base de datos y ex-

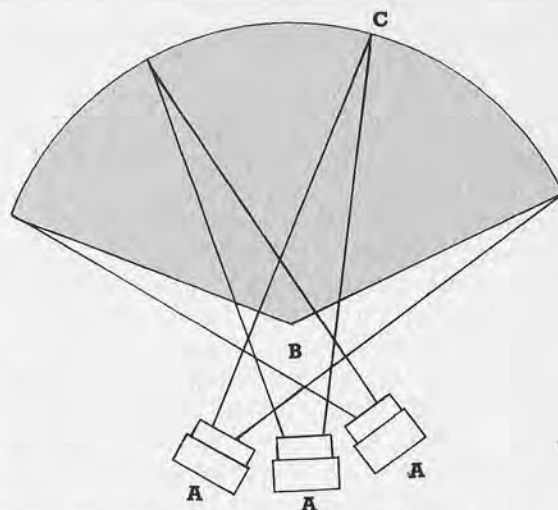


Fig. 1. Simulador de vuelo. A: conjunto de tres cámaras cada una de las cuales proyecta sobre un tercio de la pantalla. B: posición del usuario (piloto). C: pantalla de proyección de 150 grados. La zona sombreada representa el campo visual.

traer el número menor posible de elementos que puedan intervenir en la escena, clasificándolos según prioridades visuales (deben verse de delante hacia atrás). Si se desean muchos elementos en escena, la base de datos debe ser muy amplia y eficaz. Generalmente, la base de datos tiene estructura arbórea, organizada jerárquicamente por niveles de detalle. Los detalles son los polígonos, luces y otros objetos de la escena, estructurados de forma que se procese su visibilidad con rapidez y eficacia, ya que la velocidad de cómputo es fundamental.

La base de datos estructurada organiza los objetos asociándolos geográficamente en celdas o células. El procesador de estas celdas parte de un punto de vista del espectador. Aquellas celdas que no aparecen en algunos de los canales son desechadas inmediatamente.

En la base de datos de que hablamos, en el nivel más bajo, se encuentran los elementos de escena, que pueden ser polígonos o luces, cada uno con sus características especiales.

El nivel siguiente lo constituyen los objetos, que pueden contener más información adicional (aparte de la información básica que, como elementos de escena, deben aportar).

El nivel siguiente lo constituye la celda, que ya tiene un volumen y está representada en la base de datos por un nodo del árbol en que están estructurados los elementos y objetos de la escena.

Las comprobaciones de visibilidad son muy importantes, y se realizan lo más pronto

posible, para comprobar qué partes no aparecen en ningún canal de los utilizados. El manejador de objetos sólo realiza comprobaciones de tipo general dejando otras pruebas específicas para el manejador de polígonos.

El manejador de polígonos, realiza otra serie de comprobaciones de mucha mayor precisión, sobre los elementos procesados por el manejador de objetos, y almacenados en su memoria de campo. Las comprobaciones se realizan utilizando una memoria geométrica que contiene la información del punto de vista y los datos estructurales de los elementos de la escena. De todas estas comprobaciones se obtienen unos elementos que sobreviven y continúan su proceso, pasando a continuación cada elemento de la escena al procesador geométrico.

El procesador geométrico recibe los elementos de la escena (polígonos y luces) en forma de coordenadas del espacio, que se está modelizando desde el manejador de polígonos. A continuación, pasa de transformar dichos polígonos en coordenadas de canal, y a ponerlos en relación con los límites (la frontera) del canal (es decir, la porción de escena que en ese canal se va a representar). Los vértices interiores de la frontera del canal se transforman a la escala que corresponda, según la resolución de la pantalla asociada a ese canal. La descripción resultante de todos esos elementos, adecuadamente probados se almacena en la memoria de polígonos para ser procesada por el procesador de presentación.

La operación se completa mediante el procesador de presentación, que finaliza la representación de la imagen actual en el monitor. Este elemento trata con zonas rectangulares de la pantalla llamadas spans (que son simplemente un conjunto de pixels).

Existen dos posibilidades, procesar la información, analizando línea por línea, o por spans. Este último método es más rápido, y permite una resolución mayor. Las operaciones que realiza este manejador de presentación, comienzan en la memoria de polígonos, que contiene las descripciones de los objetos en un orden de prioridad visual, y que dirige los elementos de la escena al procesador de span, uno a uno, comenzando por aquel polígono que se encuentre más próximo al punto de vista. El procesador de span prepara una descripción analítica de los elementos de la escena que se encuentran en el span vigente.

Se utiliza a continuación una máscara de memoria, con el fin de borrar aquellas partes de la descripción analítica que se corresponden con partes del span al que se está aplicando la máscara en ese momento.

A continuación, se añade la descripción geométrica del polígono a la máscara de memoria, como ayuda para el proceso de escenas posteriores, dentro de ese span.

Como puede comprobarse, el nivel de exactitud al final de las etapas es enorme. La imagen es real, desde el punto de vista geométrico es fiable, sin que la división en pixels de la pantalla se vea afectada. Además, el proceso por spans permite procesar del mismo modo todas las imágenes, y, además, el filtro especial retiene detalles del dibujo.

En la actualidad se siguen mejorando estos sistemas. Algunos pretenden considerar la dirección de la mirada del piloto, por diversos sistemas, espejos, etc. Otros sistemas van provistos de un casco, que provoca rotaciones de la imagen general con el movimiento de la cabeza del piloto. Este último sistema ofrece múltiples ventajas, ya que la visión es mucho más amplia y más real.

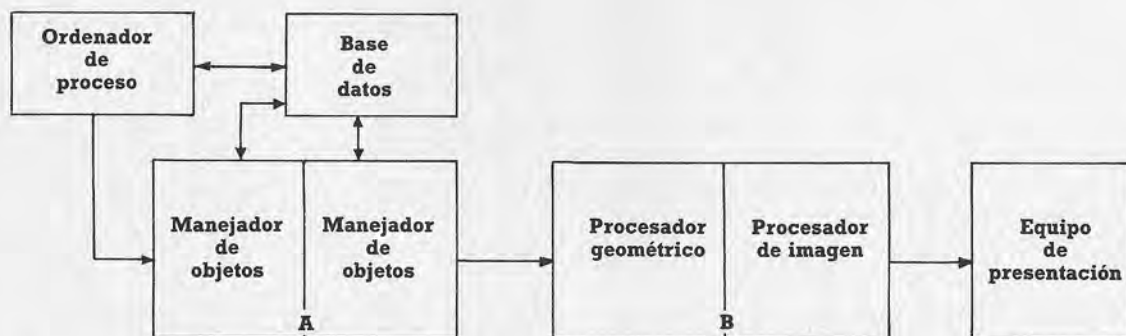


Fig. 2. Estructura genérica de un simulador. El conjunto de procesador A forma el «procesador de punto de vista» y el grupo de elementos B el «procesador de canal».

GLOSARIO DE TERMINOS UTILIZADOS EN SIMULACION

Base de Datos jerárquica. Es aquella Base de Datos en que los datos aparecen estructurados según un esquema jerárquico en forma de árbol.

Celda. Conjunto de objetos distribuidos en una disposición espacial y unidos por su relación geográfica para formar una unidad en la Base de Datos representativa de la escena.

Elemento de escena. Cada una de las figuras que componen una escena. Perfil de cada elemento del escenario a representar.

Manejador de objetos. Primer elemento del procesador de imágenes cuya misión básica es la producción de objetos necesarios para la escena, teniendo en cuenta el punto de vista del observador.

Manejador de polígonos. Depurador de los elementos producidos por el manejador de objetos, siguiendo criterios geométricos y de la estructura general de la escena a representar.

Objeto. Conjunto de elementos de escena que forman una unidad en la imagen a proyectar.

Procesador de canal. Procesador que, partiendo de los elementos generados por el procesador de punto de vista, compone la imagen y la transforma en coordenadas para el dispositivo de presentación final.

Procesador geométrico. Dispositivo que recibe los elementos de la escena y los transforma en las coordenadas necesarias para su representación, controlando los límites de la pantalla y otros datos de cada canal.

Procesador de imagen. Conjunto de dispositivos que crean la escena a representar, a partir de los parámetros introducidos y de los datos preexistentes en la Base de Datos.

Procesador de punto de vista. Dispositivo que realiza la tarea básica de extraer elementos y objetos de la Base de Datos, según la posición del observador, y se los envía al procesador de canal (a cada uno de ellos si hay varios canales y pantallas) adecuadamente preprocesados.

Span. Conjunto de pixels que se procesa como una unidad en la representación de imágenes.

VOCABULARIO DE INFORMATICA

Machine address. (Véase **Dirección absoluta.**)

Machine code. (Véase **Código máquina.**)

Machine language. (Véase **Lenguaje máquina.**)

Macroelemento. Uno o más elementos ordenados, y que se utilizan como un único elemento de datos, con un identificador único. Por ejemplo, el macroelemento «fecha» incluye el conjunto de elementos ordenados, «año», «mes» y «día».

Macroinstrucción. Instrucción, en lenguaje fuente, equivale a una secuencia específica de instrucciones máquina. Una macroinstrucción sustituye a un fragmento de programa, que puede ser repetido numerosas veces en el programa principal, mediante las adecuadas «llamadas». En ocasiones se designan simplemente por «macros».

Macroprogramación. Programación que utiliza macroinstrucciones.

Maestro. Elemento prioritario con respecto a otro. Puede referirse, por ejemplo, a un programa que controle otros programas, o a una copia de un fichero de la que se van a realizar otras copias, a un dispositivo que supervisa el trabajo de otros dispositivos, etcétera.

Maestro/esclavo. Nivel jerárquico establecido entre dos dispositivos en un entorno de comunicaciones. Si un dispositivo «supervisa» la comunicación con otro segundo, el

primero se llama maestro y el segundo pasa a ser el «esclavo» del primero. Terminada la comunicación, el nivel de jerarquía no tiene por qué mantenerse.

Magnética, bucle de histéresis. Curva cerrada, que muestra la relación existente entre la magnetización y la inducción que la produce, a lo largo de un ciclo completo.

Magnética, cinta. Cinta recubierta de una sustancia magnética sobre la que pueden almacenarse datos, polarizando selectivamente algunas de sus zonas. En una cinta magnética los datos se escriben en sentido transversal: si la cinta se escribe en nueve pistas (caso usual), en los nueve bits de las pistas (a lo ancho de la cinta) se escribe los ocho bits de un byte de datos y el bit de control de paridad).

Magnética, línea de retardo. Línea de retardo cuya operación está basada en el tiempo de propagación de ondas magnéticas.

Magnética, tarjeta. Tarjeta provista de una superficie magnética sobre la que se pueden almacenar los datos, magnetizando de forma selectiva algunas zonas. La tarjeta puede contener solamente datos grabados magnéticamente o incluir también datos observables a simple vista.

Magnética, tinta. Tinta especial que lleva una sustancia magnética detectable por sensores magnéticos.

Magnético, disco. Superficie circular plana recubierta de una sustancia magnética susceptible de ser polarizada por zonas, que sirve para almacenar datos. La densidad de grabación de los datos de una cinta se suele expresar en bpi (bytes por inch=pulgada) que significa el número de bytes (es decir, de zonas transversales de escritura de datos) que se graban en la cinta en cada pulgada (aproximadamente, 2,5 cam) de su longitud. La capacidad total de la cinta viene dada por la densidad de grabación (byts) y la longitud de cinta (800, 1600, etc., pies de longitud).

Magnético, núcleo. Material magnético, colocado de una determinada forma en relación con conductores eléctricos, y cuyas propiedades son esenciales para su utilización. Se usa para la obtención de campos magnéticos, para mantener polarizaciones en almacenamiento de datos o utilizando sus características no lineales, como elemento lógico. El material de que está compuesto suele ser hierro, óxido de hierro, ferrita, etcétera, y puede tener múltiples formas.

Magnético, tambor. Cilindro circular recubierto de una sustancia magnetizable. Se puede magnetizar por zonas para almacenar información. Los tambores magnéticos son dispositivos en los que no se alcanzan (normalmente) velocidades de acceso muy altas, pero que pueden almacenar gran cantidad de información.

Main frame. Término inglés que designa la unidad central de proceso. También puede

referirse, por extensión, a un ordenador de grandes dimensiones y potencia, en contraposición a los que se suelen llamar minordenadores.

Main program. (Véase **Programa principal.**)

Mantenimiento. Conservación del hardware y software de la máquina.

Mantenimiento preventivo. El que se realiza en un equipo, aunque no se haya producido ninguna avería, para asegurar su limpieza y su perfecto estado de conservación (y prevenir, en alguna medida, las posibles averías).

Mantisa. Parte fraccionaria del logaritmo de un número por extensión, puede referirse a la parte fraccionaria de cualquier número representado en coma flotante.

Mapa. En informática, tabla de correspondencias entre los elementos de varios conjuntos. Esquema de distribución de un área de almacenamiento de datos (memoria) en distintas zonas, para usos diversos.

Máquina. Dispositivo capaz de realizar algún trabajo. Véase tipos de máquina, etc.

Máquina, aprendizaje de la. Término utilizado en la inteligencia artificial, por el cual se alude a la capacidad de un dispositivo para mejorar sus características basándose en sus anteriores experiencias.

